HOCHSCHULE HEILBRONN
TECHNIK WIRTSCHAFT INFORMATIK

Universität Heidelberg | Hochschule Heilbronn
Studiengang Medizinische Informatik

Effiziente Differenzanalyse und Terminologie–Extraktion an Wikipedia Texten

Abschlussarbeit zur Erlangung des akademischen Grades
Diplom-Informatiker der Medizin (Dipl.-Inform. Med.)

vorgelegt von
Heiko Gramlich

Referent:	Prof. Dr. Daniel Pfeifer
Korreferent:	Prof. Dr. Martin Haag
Betreuer:	Dipl.-Inform. Med. Martin Wiesner

Abgabedatum: 08.02.2012

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen (einschließlich elektronischer Quellen) direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

(Ort, Datum)	(Unterschrift)

Danksagung

Mit Abschluss der Diplomarbeit geht ein sehr wichtiger Abschnitt in meinem Leben zu Ende. Hiermit möchte ich mich bei allen, die mir auf diesem Weg zur Seite standen, bedanken.

Zuerst möchte ich meinem betreuenden Referenten, Herrn Prof. Dr. Pfeifer danken. Er stand mir hilfreich durch Beantwortung von Unklarheiten, aber auch mit Problemlösungen zur Seite.

Ein besonderer Dank geht an Martin Wiesner, der mich während der Bearbeitung der Arbeit unterstützt hat. Durch seine intensive Betreuung, Fachwissen und Ideen konnte ich noch viel in den unterschiedlichsten Bereichen der Softwareentwicklung dazulernen. Vielen Dank, Martin!

Ein großes Dankeschön geht auch an meine Freundin, die mich gerade in der Schlussphase an Tagen, an denen nichts voranging, aufgebaut und motiviert hat.

Zu guter Letzt möchte ich meiner Familie danken! Ohne deren Unterstützung in jeglicher Hinsicht, wäre mir das Studium so sicherlich nicht möglich gewesen. Herzlichen Dank!

Abbildungsverzeichnis

Inhaltsverzeichnis

1 Einleitung

1.1 Kontext dieser Arbeit

Im Zeitalter der Informationsgesellschaft stellt das Internet eine zentrale Bedeutung für die Wissens- und Informationsbeschaffung dar. Immer mehr Menschen informieren sich mit Hilfe des Internets über das Thema Gesundheit. Gerade im Gesundheitssektor ist es wichtig, aus der großen Masse an Informationen diejenigen Quellen herauszufinden, die inhaltlich korrekt, d.h. keine Fehlinformationen enthalten, und möglichst vollständig sind, da falsche Informationen für den Nutzer sogar gesundheitsschädliche Konsequenzen haben könnten.
Als Laie medizinische Begriffe oder Zusammenhänge zwischen zwei oder mehreren Begriffen zu verstehen ist schwierig. Bei der Fülle an Informationen, die im Web angeboten werden ertrinkt der Nutzer sprichwörtlich an der Informationsflut.
Die freie Online-Enzyklopädie Wikipedia scheint in diesem Kontext eine vielversprechende Quelle zur Informationsbeschaffung zu sein. Der Gesundheitssektor der Wikipedia umfasst in der deutschen Sprache ca. 92.000 Artikel. In der englischen Sprache sind es ungefähr 350.000 Artikel.
Im Rahmen einer Diplomarbeit an der Hochschule Heilbronn, wurde von B. Trinzcek ein Framework zur Darstellung der Wikipedia als Graph entwickelt. Durch die Verlinkungen in den Artikeln ist es möglich, Zusammenhänge zwischen verschiedenen Artikeln und somit Begriffen zu visualisieren. Der durch das Framework erstellte Graph der Domäne Gesundheit wird als *Gesundheitsgraph* bezeichnet [27].
Da Verlinkung der einzelnen Wikipedia Artikel von den Autoren manuell angelegt wird, besteht die Möglichkeit, dass Zusammenhänge zwischen Artikeln existieren, welche aber nicht verlinkt worden sind und somit nicht abgebildet werden - vermeintlich strukturiertes Wissen bleibt unentdeckt.

Diese Arbeit ist einem Promotionsvorhaben von M. Wiesner an der HS Heilbronn untergeordnet, dessen Ziel es ist, ein Empfehlungssystem zu entwickeln, welches auf Basis der Krankheitsgeschichte und des Gesundheitszustands eines Patienten dem behandelnden Personal medizinische Fachartikel und dem Patienten laienverständliche Informationsmaterialien empfiehlt. Das daraus entstandene Projekt trägt den Namen TULUM[1] [33].

[1] Abkürzung für: Texbased Unique Linguistic Utilies in Medicine

1.2 Fragestellungen

Im Rahmen der Diplomarbeit soll untersucht werden, ob die Verlinkungsstrukturen der Wikipedia-Artikel im Kontext des Themenfelds Medizin und Gesundheit durch automatisierte Verfahren verbessert werden können. Hieraus leiten sich folgende Fragestellungen ab:

1. Welche Konzepte lassen sich als neue medizinisch relevante Fachterme entdecken, die bisher nicht im Gesundheitsgraph vorhanden sind, um daraus vorhandene, aber unstrukturierte Assoziationen zu erzeugen?

2. Wie lässt sich ein Verfahren zur effizienten Berechnung von neuen Konzepten in Wikipedia-Artikeln implementieren?

1.3 Ziel der Arbeit

1. Ziel dieser Diplomarbeit ist, eine Softwarelösung zu entwickeln, welche Fachterminologie aus existierenden Wikipedia-Artikeln extrahiert und daraus neue Verlinkungen zwischen den Artikeln erzeugen kann.
   Die zu entwickelnde Lösung soll dabei auf existierenden Softwarekomponenten aufbauen und die bisher automatisch generierten Gesundheitsgraphen als semantisches Netz einsetzen. Diese Datenstruktur soll um die neu entdeckten Zusammenhänge ergänzt werden.

2. Als Verfahren zur Erkennung von möglicherweise medizinisch relevanten Konzepten in vorhandenen Wikipedia-Artikeln soll die sogenannte Differenzanalyse eingesetzt werden. Diese ist im Bereich des Text-Mining etabliert und wird sowohl für die deutsche als auch für die englische Sprache durch eine Eigenimplementierung realisiert. Grundlage dafür ist sowohl ein geeigneter Referenzkorpus, als auch ein geeigneter Analysekorpus, welcher im Rahmen der Arbeit aufgebaut werden soll.

   In einer vorangegangenen Studienarbeit wurde gezeigt, dass die Differenzanalyse ein geeignetes Verfahren zur Erkennung von möglicherweise medizinischen Konzepten darstellt. In dem erarbeiteten Werkzeug wurde die Fremdbibliothek „ASV-Toolbox TE“ [29] der Universität Leipzig eingesetzt, welche diese Funktionalität übernommen hat. Es stellte sich jedoch heraus, dass diese Bibliothek nicht performant arbeitet und nur für kleine Textmengen geeignet ist [15].

Das in dieser Arbeit entwickelte Programm trägt den Namen **WikiGraph-TE-Tool**.

2 Grundlagen

In diesem Kapitel wird auf die allgemeinen und informatischen Grundlagen eingegangen, welche zum Verständnis dieser Diplomarbeit relevant sind. Dazu zählen unter anderem die Wikipedia und deren Aufbau (Kapitel 2.2) sowie die informationstechnischen Werkzeuge und Systeme (Kapitel 2.3), welche in der Arbeit eingesetzt wurden. Zuerst wird jedoch das Forschungsgebiet der Arbeit näher gebracht und grundsätzliche Begriffe definiert.

2.1 Allgemeine Grundlagen

Der Schwerpunkt der Arbeit ist dem Forschungsgebiet Text-Mining einzuordnen. Es folgen zunächst wichtige Definitionen, um Kernbegriffe der Arbeit zu klären.

2.1.1 Text-Mining

Defination nach Heyer et. al. [17]:

> Mit dem Terminus **Text-Mining** werden computergestützte Verfahren für die semantische Analyse von Texten bezeichnet, welche die automatische bzw. semiautomatische Strukturierung von Texten, insbesondere sehr großen Mengen von Texten, unterstützen.

2.1.2 Computerlinguistik

Als **Computerlinguistik** (engl: natural language processing, Abkürzung: NLP) wird das Fachgebiet bezeichnet, welches sich mit maschineller Verarbeitung natürlicher Sprache beschäftigt. Die Computerlinguistik ist im Überschneidungsbereich zwischen Informatik und Linguistik angesiedelt [10].
Das Zentrum für Informations- und Sprachverarbeitung der Universität München [11] umschreibt den Begriff folgendermaßen:

> Computerlinguistik erforscht die maschinelle Verarbeitung natürlicher Sprachen. Sie erarbeitet die theoretischen Grundlagen der Darstellung, Erkennung und Erzeugung gesprochener und geschriebener Sprache durch Maschinen. Praktische Anwendungen sind:
>
> - Klassifikation und Informationssuche in Texten (Internet-Suchmaschinen)
> - maschinelle und computerunterstützte Übersetzung
> - Umwandlung von gesprochener in geschriebene Sprache (Spracherkennung)
> - Dialogsysteme, z.B. elektronische Telefonauskunft
> - Erstellung und Nutzung sehr umfangreicher elekronischer Lexika
> - Rechtschreibkorrektur und Grammatikprüfung
> - automatische Textzusammenfassungen
> - Textgenerierung aus elektronischen Daten

2.1.3 Fachsprache und Gemeinsprache

Es folgen Erklärungen und Definitionen der Begriffe „Gemeinsprache“ und „ Fachsprache“.

Der Begriff „Gemeinsprache“ ist folgendermaßen definiert:

> Definition laut DIN2342 [22]:
> Kernbereich der Sprache, an dem alle Mitglieder einer Sprachgemeinschaft teilhaben.
>
> Definition laut Schmitz et. al. [23]:
> Unter **Gemeinsprache** versteht man den allen Mitgliedern der Sprachgemeinschaft verständlichen (passiven) und von ihnen benutzten (aktiven) Kernbereich der Sprache .

Die Gemeinsprache kann demnach also für jeden Menschen, der einer Sprache mächtig ist, angewandt und verstanden werden, ohne Fachkenntnis über ein Gebiet zu besitzen.

Der Begriff „Fachsprache“ gemäß DIN 2342 [22] definiert sich wie folgt:

> **Fachsprache** :
> Bereich der Sprache, der auf eindeutige und widerspruchsfreie Kommunikation in einem Fachgebiet gerichtet ist und dessen Funktionieren durch eine festgelegte Terminologie entscheidend unterstützt wird.

Um die obige Definition verständlicher zu machen, wird noch der Begriff „Terminologie“ definiert:

> **Terminologie** (auch: Fachwortschatz)
> Gesamtbestand der Begriffe und ihrer Benennungen in einem Fachgebiet. [22]

Laut Fluck [14] ist eine Fachsprache eine „... häufig verwendete Bezeichnung, die alle möglichen, verbalen und nichtverbalen [z.B. Symbole, Formeln, Graphiken] (Text-)Formen der fachbezogenen Verständigung (Fachkommunikation, Fachdiskurs) meint und oft in einem Gegensatz zum Begriff der üblicherweise verwendeten „Gemeinsprache”... gestellt wird.“
Gemäß Möhn bilden Fachsprachen „innerhalb der Gesamtsprache auf einzelne Fachgebiete bezogene, in sich differenzierte Subsysteme, die durch eine charakteristische Auswahl, Verwendung und Frequenz sprachlicher Mittel definiert sind.“ (als Zitat entnommen aus [23])

Abgrenzung Gemeinsprache von Fachsprache

Eine genaue Grenze zwischen Gemein- und Fachsprache lässt sich nicht ziehen, da eine Fachsprache auf dem Gerüst der Gemeinsprache aufgebaut ist. Laut Heyer et. al [17] unterscheidet sich eine Fachsprache in folgenden Merkmalen von der Gemeinsprache:

- Von Gemeinsprache abweichende Grammatik
- Hohe Wahrscheinlichkeit bestimmter syntaktischer Konstruktionen
- Fachterminologie
- Charakteristische Morpheme, wie z.B. *-itis

In Abbildung 2.1 ist der Zusammenhang von Allgemein- und Fachsprache verdeutlicht.

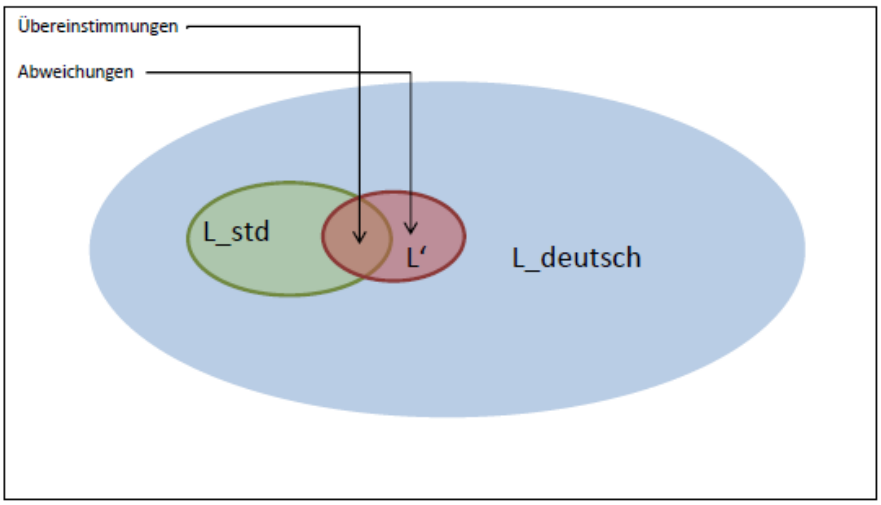


Abbildung 2.1: Verdeutlichung des Verhältnises zwischen Gemein- und Fachsprache

Dabei bezeichnet L_deutsch die Menge der syntaktisch wohlgeformten Sätze des Deutschen. Die Menge der Sätze der deutschen Standartsprache (Gemeinsprache) L_std, welche in der Praxis durch eine Kollektion deutschsprachiger Texte repräsentiert wird, ist eine Teilmenge hiervon. Die Fachsprache L' ist ebenfalls eine Teilmenge von L_deutsch. Je nachdem wie die Fachsprache von der Gemeinsprache abweicht, variiert der Umfang der Schnittmenge zwischen den Teilmengen L' und L_std [17].

Definition des Begriffs „Terminologie-Extraktion“ nach G. Heyer et al. [17]:

> Ziel der **Terminologie-Extraktion** ist es, das domänenspezifische Vokabular eines Fachgebiets anhand von Fachtexten herauszufinden.

2.1.4 Zipfsche Gesetze

Die Häufigkeit von Wortfmormen spielt in linguistischen Untersuchungen eine wesentliche Rolle. Oftmals können Ergebnisse nur dann verglichen werden, wenn diese auf Wortformen basieren, welche in etwa gleich häufig in den zu untersuchenden Texten auftreten. Als Textkorpora werden im Folgenden die Resourcen des Projekts „Deutscher Wortschatz“ der Universität Leipzig, sowohl für die deutsche, als auch für die englische Sprache genutzt. Beide Korpora beinhalten 10 Millionen Sätze, welche aus Zeitungsartikeln extrahiert wurden [30].
Ordnet man die in einem Text enthaltenen Wortformen absteigend nach deren absoluter Häufigkeit in einer Liste an (vgl. Tabelle 2.1 und Tabelle 2.2), kann eine Gesetzmäßigkeit erkannt werden [17].

Rang n	Wortform	Häufigkeit n	$r * n$
1	die	4.280.255	4.280.255
2	der	4.169.928	8.339.856
3	und	3.250.772	9.752.316
4	in	2.360.535	9.442.140
5	den	1.655.584	8.277.920
6	ist	1.528.544	9.171.264
7	zu	1.463.917	10.247.419
8	das	1.441.476	11.531.808
9	nicht	1.297.392	11.676.528
10	von	1.258.578	12.585.780

Tabelle 2.1: Deutsche Wortformen mit Sortierung nach Rang

Rang n	Wortform	Häufigkeit n	$r * n$
1	the	9.294.217	9.294.217
2	to	4.969.920	9.939.840
3	and	4.346.865	13.040.595
4	a	4.213.139	18.525.566
5	of	4.059.051	20.295.255
6	in	3.237.641	19.425.846
7	is	1.885.768	13.200.376
8	that	1.873.579	14.988.632
9	for	1.810.527	16.294.743
10	on	1.424.292	14.242.920

Tabelle 2.2: Englische Wortformen mit Sortierung nach Rang

Der Rang r einer Wortform multipliziert mit deren Häufigkeit n ist in etwa konstant, das heißt mit textabhängiger Konstante k:

$$r * n \approx k \tag{2.1}$$

Die Auftretenshäufigkeit n einer Wortform ist also umgekehrt proportional zu ihrem Rang r.

$$n \sim 1/r \tag{2.2}$$

Tabelle 2.3 zeigt einen gröberen Ausschnitt der Worthäufigkeiten im deutschen Wortschatz.

Rang n	Wortform	Häufigkeit n	$r * n$
10	von	1.258.578	12.585.780
100	alle	147.673	14.767.300
500	führt	26.276	13.138.000
1000	Freunde	13.999	13.999.000
5000	bayerischen	2590	12.950.000
10000	Parallel	1116	11.160.000

Tabelle 2.3: Deutsche Wortformen mit Sortierung nach Rang

Hier wird der Zusammenhang zwischen Häufigkeit n und Rang r (vgl. Formel 2.1) deutlicher. Das Produkt aus n und r ist in etwa konstant. In Abbildung 2.2 werden Rang r und Häufigkeit n mit logarithmischen Achsen gegenüber gestellt. Die Grafik zeigt, dass die Gerade im mittleren Frequenzbereich der Wortformen, die Formel von Zipf, also den Zusammenhang von Rang und Häufigkeit, relativ gut beschreibt.

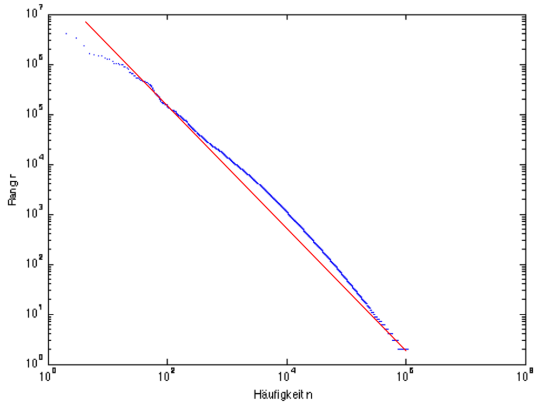


Abbildung 2.2: Deutsche Wortformen: Rang gegenüber Häufigkeit

In Abbildung 2.3 ist die Ranglistenverteilung der ersten 250 Wortformen dargestellt.Es ist erkennbar, dass die Worthäufigkeiten ungleich verteilt sind. Das Zipfsche Gesetz lässt sich also folgendermaßen interpretieren: Es gibt wenige Wortformen, welche häufig vorkommen, aber viele Wortformen, die sehr selten in der Allgemeinsprache vorkommen.

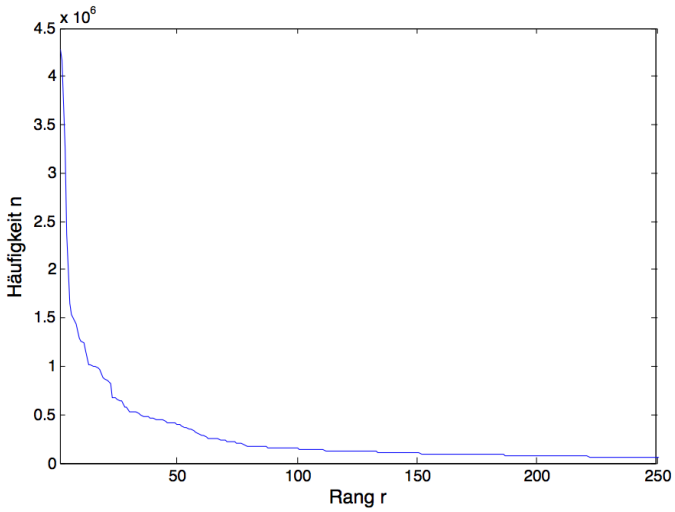


Abbildung 2.3: Deutsche Wortformen: Ranglistenverteilung der ersten 250 Wortformen

Die Zipfsche Gesetze werden zur Bestimmung der sogenannten Häufigkeitsklassen verwendet, auf welche im nächsetn Kapitel eingegangen wird.

2.1.5 Häufigkeitsklasse

Die Häufigkeitsklasse ist ein statistisches Maß für die Gebrauchshäufigkeit einer Wortform in einer natürlichen Sprache oder einem Sprachausschnitt [38].
Die Berechnung der Häufigkeitsklasse einer Wortform w wird mit folgender Formel berechnet [17]:

$$HKL(w) = ganzerAnteil(\log_2\left(\frac{|h|}{|w|}\right)) \tag{2.3}$$

Dabei ist $|w|$ die Auftrittshäufigkeit der Wortform w und h die Häufigkeit des am häufigsten vorkommenden Wort des Referenzkorpus.
In der deutschen Sprache gilt $h =$ '*die*', in der englischen Sprache gilt $h =$ '*the*'.
Das beudeutet, dass die Wortform h ungefähr $2^{HKL(w)}$-mal so häufig auftritt als w [17].
Das häufigste Wort einer Sprache wird der Häufigkeitsklasse 0 zugeordnet und ist im Allgemeinen das einzige Element [38]. Für die medizinischen Fachterms „Patient“ in der deutschen Sprache bzw. „patient“ in der englischen Sprache ergeben sich folgende Häufigkeitsklassen:

$$HKL('Patient') = 11$$
$$HKL('patient') = 16$$

Die Häufigkeitsklassen der Wortformen einer Sprache können auch per Zipfschem Gesetz (vgl. Kapitel 2.1.4) berechnet werden, indem die Ranglistenverteilung der Wortformen von Rangplatz $2^{i-1} + 1$ bis 2^i in die Häufigkeitsklasse i einordnet wird.
Die Häufigkeitsklasse eines Terms wird im Schritt der Terminologie-Extraktion (vgl. Kapitel 3.1.2 berechnet.

2.1.6 Part-Of-Speech-Tagging

Das Part-Of-Speech-Tagging (Tag (engl. = „Etikett“)), kurz POS-Tagging, ist ein Verfahren in der Computerlinguistik, bei dem den Wortformen eines Satzes ihre jeweilige Wortart maschinell zugeordnet wird. Diese Markierungen der einzelnen Wortformen erfolgt durch ein sogenanntes Tagset (Set (engl. = „Menge“)), einer Menge von Markierungen, bei der jede einzelne Markierung für eine Wortart steht, vorgenommen. In dem in der Arbeit implementierten Programm wird das Stuttgarter/Tübinger Tagset (kurz: STTS) verwendet [31].
Damit ist es möglich, alle Informationen eines Satzes inhaltlich erfassen zu können.

Als wichtigste Wortarten des deutschen Wortschatzes werden folgende bezeichnet [32]:

- Substantiv (Hauptwort, Nomen)
- Adverb (Eigenschaftswort)
- Verb (Tätigkeitswort)
- Präpositionen (Verhältniswort)
- Konjunktionen (Bindewort)

Abbildung 2.4 zeigt einen Satz in Normalform und in der mit dem STTS-Tagset markierten Darstellung.

Normalform:
„Der Patient wurde sofort ins Krankenhaus eingeliefert.“

POS-Tagging:
„Der[ART] Patient[NN] wurde[VAFIN] sofort[ADV] ins[APPRART] Krankenhaus[NN] eingeliefert[VVPP].[$.]“

Abbildung 2.4: Beispiel zu Part-of-Speech-Tagging mit STTS-Tagset

Im Rahmen dieser Arbeit wird ein POS-Tagger verwendet, um potentielle Fachterme anhand bestimmten POS-Mustern zu erkennen, wie z.B. „grüner Star“: [ADJA][NN]. In Kapitel 2.3.2.3 wird genauer auf den eingesetzten POS-Tagger eingegangen.

2.2 Wikipedia und deren Aufbau

Die Wikipedia [36] dient als Wissensbasis fur TULUM. Aus diesem Grund wird im Folgenden, zum näheren Verstandnis der Wikipedia, deren Aufbau aufgezeigt.

Die Wikipedia ist ein freies Online-Lexikon, welches 2001 gegründet wurde. Der Name „Wikipedia“ setzt sich aus „Wiki“ (hawaiisch für „schnell“) und „Encyclopedia“ (dem englischen Wort für Enzyklopädie) zusammen [34]. Die Wikipedia umfasst aktuell 283 verschiedene Sprachen. Die umfangreichste Sprachversion ist die englische mit 3,84 Millionen Artikeln, gefolgt von der deutschen Version mit 1,34 Millionen Artikeln [35].

Die gesamten Inhalte der Wikipedia stehen unter freien Lizenzen. Alle Artikeltexte stehen unter der GNU-Lizenz für freie Dokumentation; seit dem 15. Juni 2009 auch unter der Creative- Commons-Attribution-ShareAlike-Lizenz (CC-BY-SA[1]). Somit können (unter bestimmten Bedingungen) die Artikel selbst kommerziell genutzt, verändert und verbreitet werden.

Auf die Nutzung der Wikipedia wird in Kapitel 2.3.4 näher eingegangen.

2.2.1 Wikipedia-Artikel

Die Wikipedia besteht aus vielen verschiedenen Webseiten, welche als sogenannte „Namensräume“ in Gruppen zusammengefasst sind. Der wichtigste Namensraum ist der Namensraum „Artikel“, welcher die enzyklopädischen Artikel beinhaltet. Jeder Artikel behandelt ein einzelnes Konzept und ist eindeutig benannt.

Beispiele für Wikipedia-Artikel:

- „Patient“
- „Mittelhandknochen“
- „Krankenhaus“

Ein weiterer Namensraum, welcher für diese Arbeit wichtig ist, ist der Namensraum „Kategorie“. Jeder Artikel ist einer oder mehreren Kategorien zugeteilt.

[1] http://creativecommons.org/

In Abbildung 2.5 ist ein Screenshot des Wikipedia-Artikels „Patient“ zu sehen.

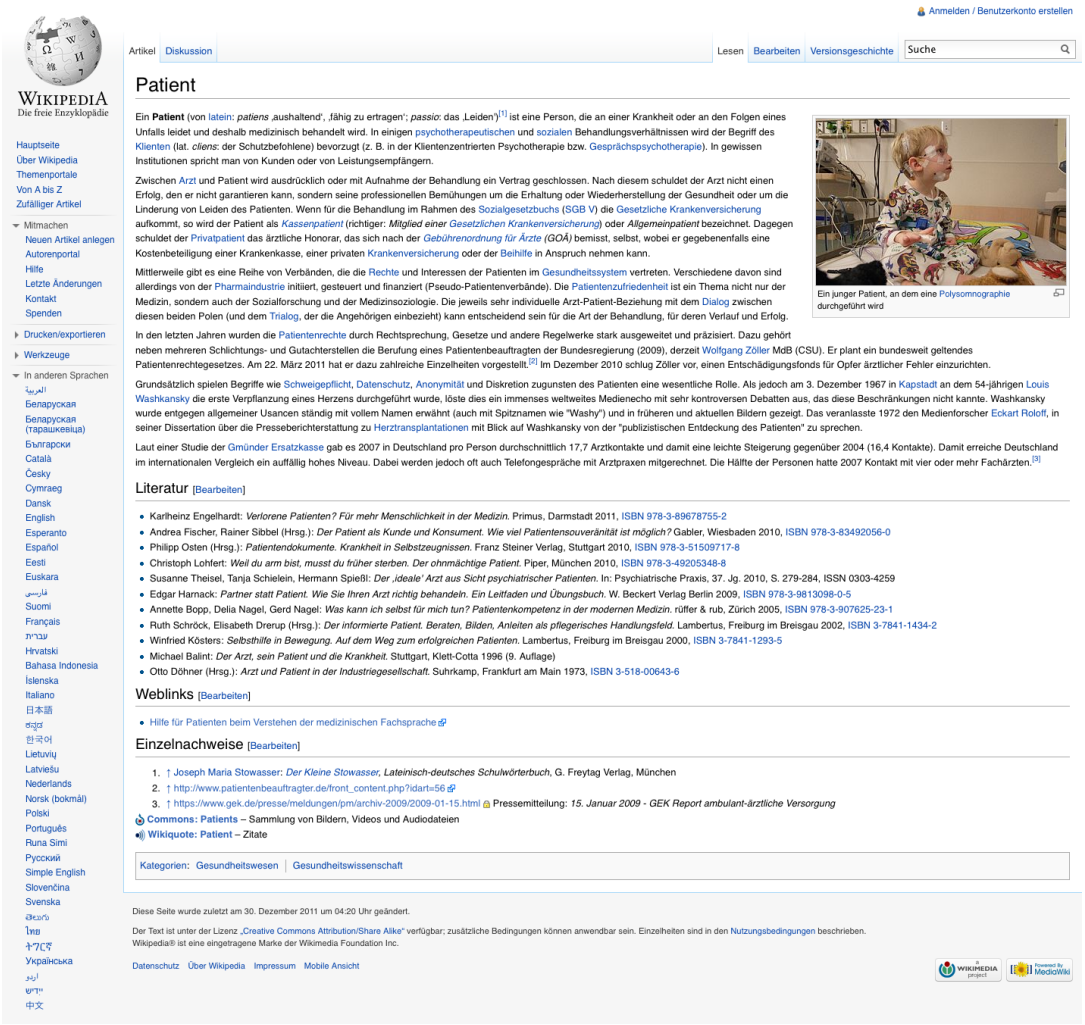


Abbildung 2.5: Screenshot: Wikipedia-Artikel „Patient“

Ein Artikel setzt sich größten Teils aus Fließtext zusammen. Der Text selbst behandelt das Thema des Konzepts. Die Texte sind meist in verschiedene Absätze gegliedert. Sie besitzen Überschriften oder sind durch sonstige Formatierungen hervorgehoben. Artikel können durch zusätzliches Medienmaterial, zum Beispiel Bilder, Videos oder Audioaufnahmen angereichert sein.
Anstatt HTML wird zur Formatierung von Wikipedia-Artikeln eine vereinfachte alternative Auszeichnungssprache verwendet, welche *Wiki-Text*, *Wiki-Syntax*, *Wiki-Code* oder *wiki markup* genannt wird. Dies hat den Vorteil, dass dadurch das Verfassen von Artikeln vereinfacht und übersichtlicher dargestellt werden kann [37]. Im Folgenden wird die Bezeichnung *wiki markup* verwendet.
Abbildung 2.6 zeigt die benötigte Codezeile zur Verlinkung zum Wikipedia-Artikel

„Hyperlink“ mit Hilfe von HTML.

```
<a href="/wiki/Hyperlink">Hyperlink</a>
```

Abbildung 2.6: Erzeugung einer Verlinkung zum Wikipedia Artikel „Hyperlink“ mit Hilfe von HTML

In Abbildung 2.7 wird in *wiki markup* ein Link zum Wikipedia-Artikel „Hyperlink“ erzeugt.

```
[[Hyperlink]]
```

Abbildung 2.7: Erzeugung einer Verlinkung zum Wikipedia-Artikel „Hyperlink“ mit Hilfe von wiki markup

2.2.1.1 Artikel-Links

Mit Hilfe von sogenannten „Artikel-Links“ ist es möglich, innerhalb eines Artikels auf Inhalte der Wikipedia zu verlinken.
In Abbildung 2.8 ist die Verlinkung aus einem Artikel auf einen anderen Artikel via *wiki markup* dargestellt.

```
Zwischen [[Arzt]] und Patient wird ausdrücklich oder mit Aufnahme der Behandlung ein Vertrag geschlossen.
```

Abbildung 2.8: Verlinkung zum Wikipedia-Artikel „Arzt“ aus dem Artikel „Patient“ mit Hilfe von wiki markup

Abbildung 2.9 zeigt, wie die zuvor erzeugte Verlinkung in einem Internet-Browser dargestellt wird. Die Verlinkung ist farblich hervorgehoben.

Abbildung 2.9: Artikel-Link auf den Artikel „Arzt“ aus dem Artikel „Patient“ in Browserdarstellung

2.2.1.2 Kategorie-Links

Am Ende eines Artikels befindet sich meist ein Bereich mit der Information, in welcher Kategorie der aktuelle Artikel geführt wird. Ein Artikel kann mehreren Kategorien angehören. Abbildung 2.10 zeigt ein Screenshot mit den Kategorien, in welchen der Artikel „Patient“ verwaltet wird.

Abbildung 2.10: Kategorien, in welchen der Artikel „Patient“ verwaltet wird (in Browserdarstellung)

Diese Information wird folgendermaßen mit Hilfe von *wiki markup* realisiert (vgl. Abb. 2.11).

```
[[Kategorie:Gesundheitswesen]] [[Kategorie:Gesundheitswissenschaft]]
```

Abbildung 2.11: Kategorien, in welchen der Artikel „Patient“ verwaltet wird (via wiki markup)

Mit Hilfe der Kategorie-Links können Verbindungen zwischen Artikel und den Kategorien, in welchen der Artikel behandelt wird, hergestellt werden. Anhand der Kategorie wird ein Artikel dem WikiGraphen hinzugefügt [27]; Kategorie-Links werden daher nur indirekt im Rahmen dieser Arbeit genutzt.

2.3 Informatische Grundlagen

Die informatischen Grundlagen dieser Arbeit umfassen verschiedene Systeme und Werkzeuge, die zur Implementierung des WikiGraph-TE-Tools verwendet wurden. Dazu zählen das Build-Management-Werkzeug Maven (Kapitel 2.3.1), verschiedene Werkzeuge aus dem Bereich der Computerlinguistik (Kapitel 2.3.2 und Kapitel 2.3.3), sowie ein Framework zur Nutzung der Wikipedia (Kapitel 2.3.4).

2.3.1 Apache Maven

Maven ist ein modulisiertes Build-Management-Tool, welches von der Apache Foundation entwickelt wurde [1]. Maven basiert auf Java und wird zur professionellen Softwareentwicklung genutzt. Der Name Maven stammt aus dem Jiddischen und kann mit „Sammler des Wissens“ übersetzt werden.
Maven bietet die Funktionalität jedes Java-basierte Softwareprojekt zu managen. Dazu verwendet das Tool selbst definierte „Lifecycles“, wie zum Beispiel das automatisierte Zusammenbauen des aktuellen Softwareprojekts, die automatische Generation der Projektdokumentation, das integrierte Testen oder das Bereitstellen des aktuellen Projekts als ausführbare Datei.
Neben den bisher genannten Funktionalitäten ist das Einbinden von sogenannten „Maven-Dependencies“ (engl. Abhängigkeiten) ein weiteres Feature von Maven. Dadurch ist es möglich, selbst entwickelte Softwarebibliotheken oder Fremdbibliotheken ohne großen Aufwand in das eigene Softwareprojekt einzubinden. Dies geschieht über ein zentrales Repository[2], welches alle verfügbaren Dependencies beinhaltet. Erfolgt die Auswahl einer Dependency, wird diese automatisch zur Verfügung gestellt und in ein lokales Repository übertragen.
Typische Beispiele für eingebundene Dependencies sind *JUnit*, zum Testen von Java-Programmen, oder *log4J* als Logging-Tool.
Die einzelnen entwickelten Module und Komponenten von TULUM werden ausschließlich als Maven-Projekte implementiert, um eine einfache Modularisierung sowie Integration in ein Gesamtprojekt zu gewährleisten.

[2]Definition: verwaltetes Verzeichnis zur Speicherung und Beschreibung von digitalen Objekten [24]

2.3.2 Apache OpenNLP

Apache OpenNLP ist eine Zusammenstellung von Werkzeugen, welche im Bereich der Computerlinguistik (vgl. Kapitel 2.1.2) angesiedelt sind. Die einzelnen Werkzeuge sind in Java programmiert und Open Source (quelloffen) [2]. OpenNLP kann per Maven-Dependency (Kapitel 2.3.1) leicht in ein Projekt eingebunden werden. Jedes Werkzeug ist dank Internationalisierung für verschiedene Sprachen einsetzbar. Um ein Werkzeug nutzen zu können, muss eine binäre „Model"-Datei, welche spezifisch für jede Sprache ist, während der Initialisierungsphase in das Programm geladen werden [4].

Auf die im WikiGraph-TE-Tool verwendeten Werkzeuge wird im Folgenden eingegangen.

2.3.2.1 Sentence Detector

Der Sentence Detector dient der Zerlegung eines Textes in einzelne Sätze. Dabei kann das Werkzeug erkennen, ob ein Satzzeichen das Ende eines Satzes markiert oder nicht [3].

Abbildung 2.12 zeigt ein Beispiel zur Funktionsweise des Sentence Detectors: Es sind drei Sätze hintereinander geschrieben. Dem menschlichen Auge fällt es schwer, die einzelnen Sätze, ohne mehrmaliges Lesen, zu entwirren.

> Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29. Mr. Vinken is chairman of Elsevier N.V., the Dutch publishing group. Rudolph Agnew, 55 years old and former chairman of Consolidated Gold Fields PLC, was named a director of this British industrial conglomerate.

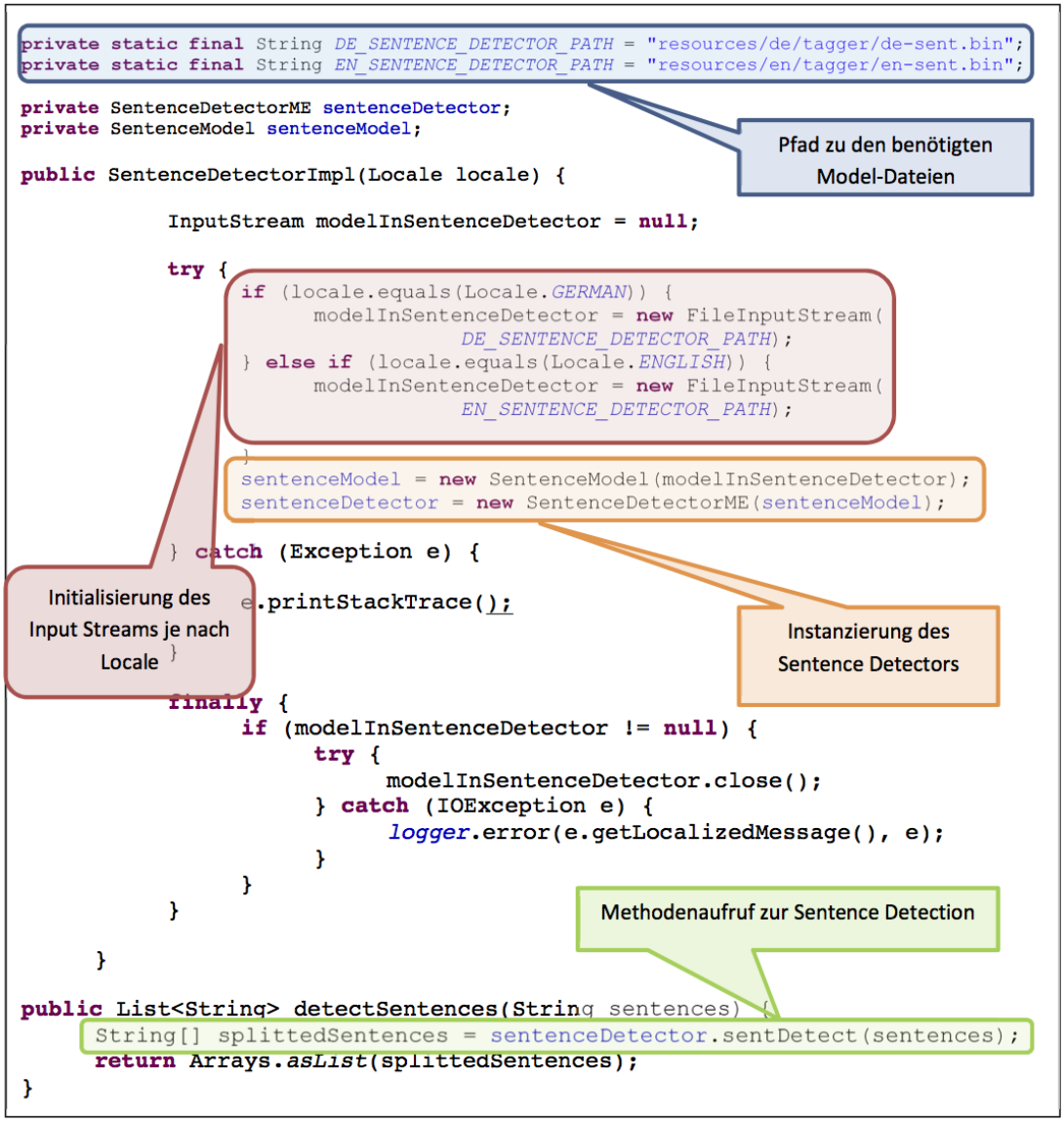
Abbildung 2.12: Beispieltext vor Anwendung des Sentence Detector

Abbildung 2.13 stellt die Sätze nach durchgeführter Satzerkennung dar (jeder Satz ist durch eine Leerzeile voneinander getrennt).

> Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.
>
> Mr. Vinken is chairman of Elsevier N.V., the Dutch publishing group.
>
> Rudolph Agnew, 55 years old and former chairman of Consolidated Gold Fields PLC, was named a director of this British industrial conglomerat

Abbildung 2.13: Beispieltext nach Anwendung des Sentence Detector

Die Einbindung und Nutzung des Sentence Detectors in eine eigene Applikation ist in Abbildung 2.14 dargestellt.




```
private static final String DE_SENTENCE_DETECTOR_PATH = "resources/de/tagger/de-sent.bin";
private static final String EN_SENTENCE_DETECTOR_PATH = "resources/en/tagger/en-sent.bin";

private SentenceDetectorME sentenceDetector;
private SentenceModel sentenceModel;

public SentenceDetectorImpl(Locale locale) {

        InputStream modelInSentenceDetector = null;

        try {
            if (locale.equals(Locale.GERMAN)) {
                modelInSentenceDetector = new FileInputStream(
                        DE_SENTENCE_DETECTOR_PATH);
            } else if (locale.equals(Locale.ENGLISH)) {
                modelInSentenceDetector = new FileInputStream(
                        EN_SENTENCE_DETECTOR_PATH);
            }
            sentenceModel = new SentenceModel(modelInSentenceDetector);
            sentenceDetector = new SentenceDetectorME(sentenceModel);

        } catch (Exception e) {
            e.printStackTrace();
        }
        finally {
            if (modelInSentenceDetector != null) {
                try {
                    modelInSentenceDetector.close();
                } catch (IOException e) {
                    logger.error(e.getLocalizedMessage(), e);
                }
            }
        }

    }

public List<String> detectSentences(String sentences) {
    String[] splittedSentences = sentenceDetector.sentDetect(sentences);
    return Arrays.asList(splittedSentences);
}
```

Abbildung 2.14: Codeausschnitt: SentenceDetectorImpl.java

Je nach gewünschter Sprache wir eine andere Model-Datei bei der Initialisierung des *sentenceModel* in das Programm geladen. Das erzeugte *sentenceModel* wird danach dem *sentenceDetector* als Parameter übergeben. Über die *sentenceDetector*-Instanz kann nun die Satzerkennung aufgerufen werden, welche ein String-Array der gefundenen Sätze zurückliefert.

2.3.2.2 Tokenizer

Der Tokenizer dient zur Zerlegung eines Satzes in die einzelne Wörter und Satzzeichen. Dies ist ein zweistufiger Prozess: zuerst werden Satzgrenzen (wie „!“, „?“ oder „.“) ermittelt, danach werden die einzelnen Elemente des Satzes identifiziert [3].
Abbildung 2.15 zeigt einen Beispielsatz vor und nach Anwendung des Tokenizers.

vor Anwendung des Tokenizers:
Mr. Vinken is chairman of Elsevier N.V., the Dutch publishing group.

nach Anwendung des Tokenizers: (durch Semikolon getrennt)
Mr;.;Vinken;is;chairman;of;Elsevier;N;.;V;.;,;the;Dutch;publishing;group;.

Abbildung 2.15: Beispieltext vor und nach der Anwendung des Tokenizers

Die Einbindung des Tokenizers in ein Programm erfolgt analog zur Einbindung des Sentence Detectors. Je nach gewünschter Sprache wird eine andere Model-Datei in den Speicher geladen und dann einer Tokenizer-Instanz als Parameter übergeben (vgl. Abbildung 2.16).

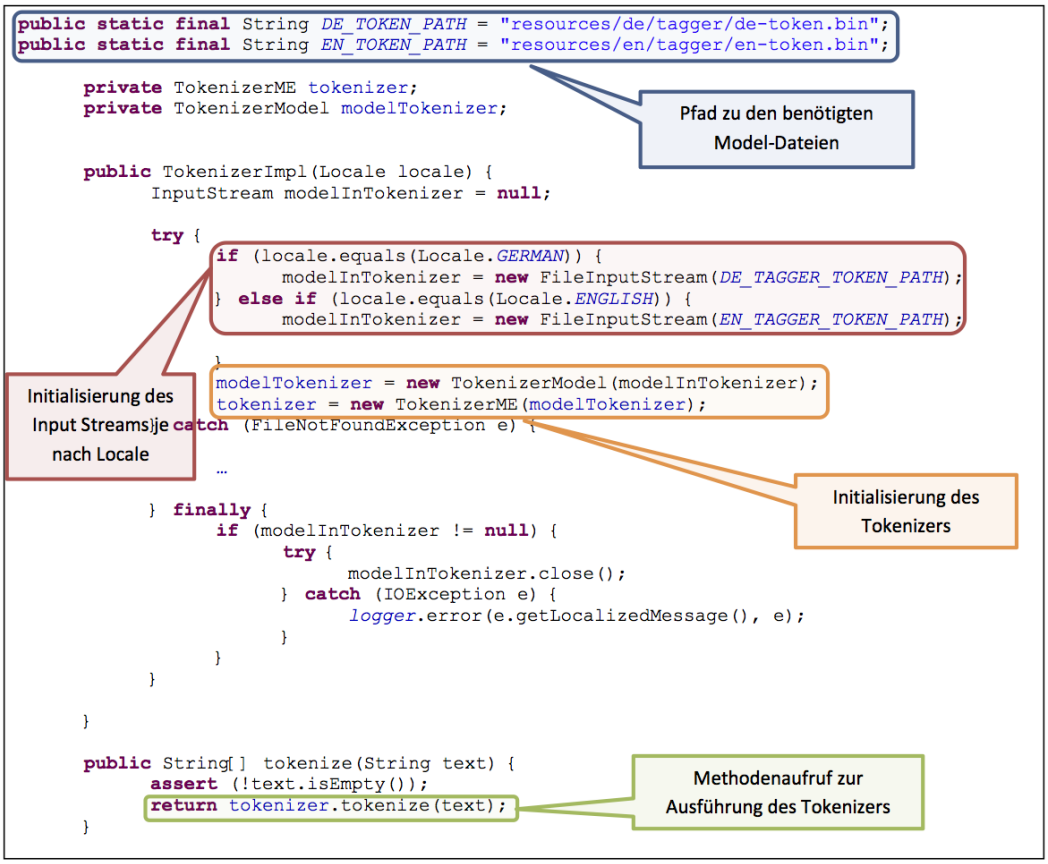


Abbildung 2.16: Codeausschnitt: TokenizerImpl.java

2.3.2.3 POS-Tagger

Wie schon in Kapitel 2.1.6 dargestellt, wird der POS-Tagger eingesetzt, um einzelnen Satzkomponenten deren Wortart zuzuteilen.
Die Einbindung des POS-Taggers ist in Abbildung 2.17 dargestellt. Auch bei dieser Softwarekomponente wird, je nach gewünschter Sprache, eine spezifische Model-Datei eingelesen und daraus die *tagger*-Instanz erzeugt. Im Gegensatz zu den bisher besprochenen Tools des OpenNLP-Pakets benötigt der POS-Tagger die Vorverarbeitung des zu „taggenden“ Textes per Tokenizer. Nachdem der zu „taggende“ Text in seine Satzteile zerlegt wurde, werden die Tokens der *tag()*-Methode der *tagger*-Instanz übergeben. Nachdem den einzelnen Satzteilen ihre Wortart zugeteilt wurde, werden die Wahrscheinlichkeiten für die korrekte Wortart-Vergabe durch die Methode *probs()*, welche bei jedem Tagging-Vorgang berechnet werden, geliefert. Um eine möglichst hohe Prezission während der weiteren Verarbeitung zu gewährleisten, werden nur diejenigen Satzteile in die Ergebnismenge übernommen, bei denen die Wahrscheinlichkeit für eine korrekte Vergabe des Tags größer/gleich *PROPABILITY_TAGGER* ist.

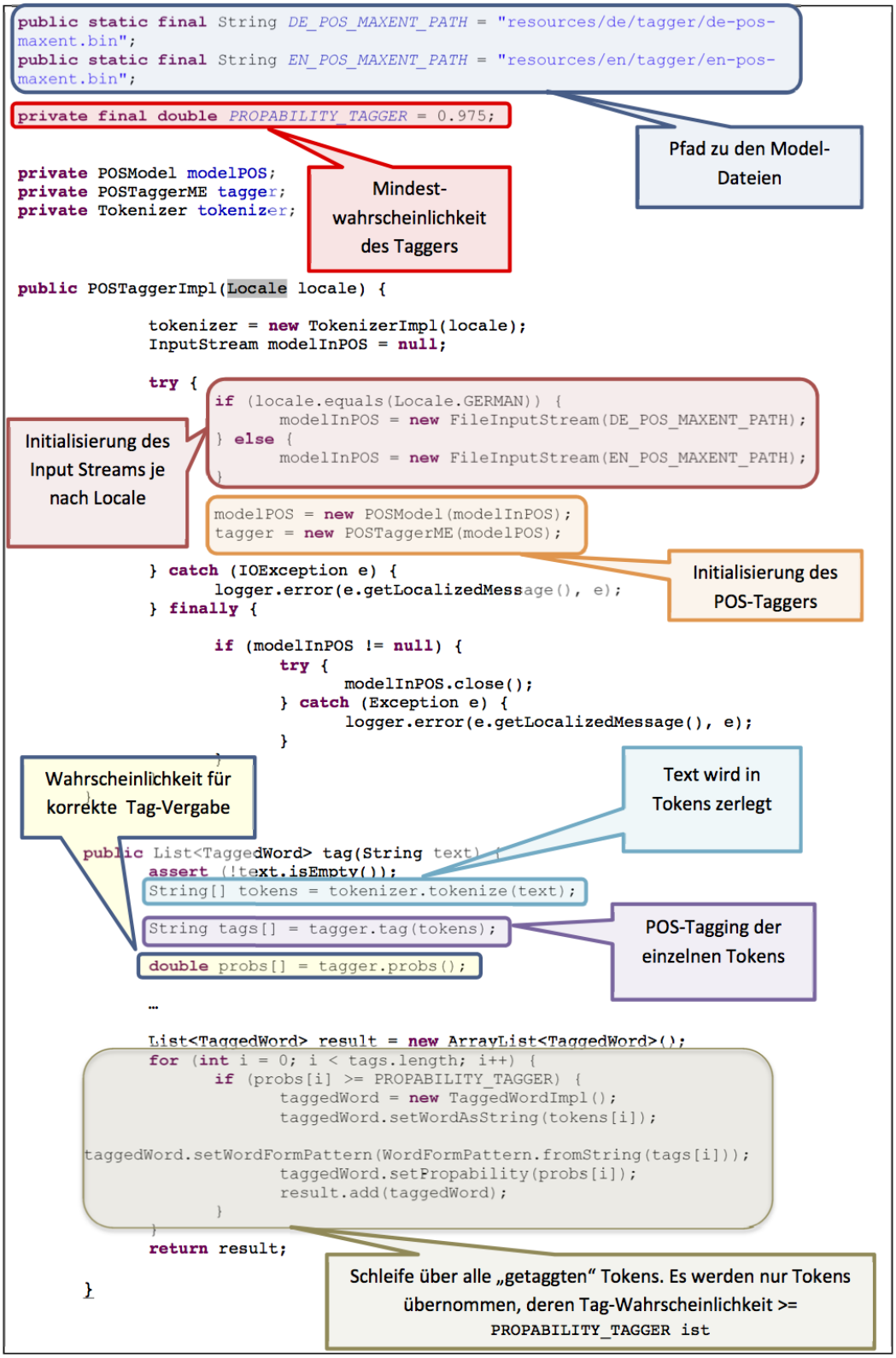


Abbildung 2.17: Codeausschnitt: POSTaggerImpl.java

2.3.3 Grundformreduktion

Die Grundformreduktion (engl. stemming) wird in IR-Systemen eingesetzt, um morphologische Varianten eines Worts auf deren gemeinsamen Wortstamm zurückzuführen. Der meist verbreitete Stemmer ist der sog. „Porter-Stemmer“, welcher nach seinem Schöpfer Martin Porter benannt ist. Es handelt sich hierbei um den Quasi-Standart der Stemming-Verfahren für die englische Sprache. Dabei wird ein Term nach gewissen Regeln Schritt für Schritt auf seinen Wortstamm reduziert [20].
Nachfolgendes Beispiel zeigt die Arbeitsweise des Algorithmus. Dabei werden die entstandenen Ergebnisse der Teilschritte anhand des Terms „generalizations“ aufzeigt:

„generalizations“ → „generalization“ → „generalize“ → „general“ → “gener“

Auf Grund der Reduzierung der Terme auf deren Wortstamm und nicht auf deren Grundform, wie in diesem Fall „generalization“, ist dieses Verfahren im Kontext dieser Arbeit ungeeignet.

Eine vielversprechende Lösung stellt dagegen die Lemmatisierung dar. Die Lemmatisierung ist ein Verfahren, welches ein Wort auf das sogenannte Lemma zurückführt, eine Grundform, wie sie beispielsweise in Lexika verwendet wird [42].
Beispiele hierfür sind:

- *Häuser* → *Haus*
- *Stati* → *Status*
- *Noten* → *Note*

Im Englischen ist die Lemmatisierung im Wesentlichen das Streichen von Suffixen (engl. „suffix stripping“) unter Beachtung von gewissen Regeln [42]. Die Pluralbildung folgt neun Regeln sowie einer geringen Anzahl von ungefähr hundert Ausnahmen [25]. Beispiele für die Regeln zur Pluralbildung im Englischen sind:

- nach Zischlauten wird -es angehängt: *a box* → *two boxes*
- y nach Konsonant wird zu -i, dann wird -es angehängt: *a city* → *two cities*
- y nach Vokal bleibt: *a boy* → *two boys*

Der für die englische Sprache eingesetzte Lemmatizer ist eine Eigenimplementierung, welche die Regeln der Pluralbildung umgekehrt anwendet. Des Weiteren wird mit einer Ausnahmeliste gearbeitet, welche irreguläre Pluralbildungen enthält. Der Quellcode kann in Anhang A1 der beigefügten CD eingesehen werden.

In der deutschen Sprache sind acht verschiedene Pluaralendungen vorhanden, jedoch lässt sich keine allgemeingültige Regel zur Pluralbildung bestimmen. Die meisten Substantive bilden ihren Plural mit “(e)n“ oder “e„ beispielsweise *Note* → *Noten*.
Einige Substantive bilden ihre Pluralform mit Umlaut (ä,ö,ü), wie zum Beispiel *Haus*

→ *Häuser* [6]. Die Bildung des Plurals per Umlaut und das sogenannte „Genitiv-S“, zum Beispiel *“des Hauses„* oder *“des Bootes’*, erschwert die automatisierte regelbasierte Grundformreduktion erheblich. Daher werden im Deutschen normalerweise lexikonbasierte Ansätze anstatt Umschreiberegeln angewandt [42].
Der im WikiGraph-TE-Tool eingesetzte Lemmatizer für die deutsche Sprache basiert auf dem „Durm German Lemmatizer“ [41]. Das darin enthaltene Lexikon, welches 84.320 Einträge beinhaltet, wurde aus dem NEGRA Korpus[3] erstellt[19]. Da nach ersten Trainingsläufen Fehler bei der Lemmatisierung festgestellt wurden, erfolgte die komplette manuelle Überarbeitung des Lexikons. Um eine größere Anzahl an Einträgen zu erhalten, wurde das überarbeitete Lexikon mit dem, in TULUM verwendeten, „beolingus-Dictionary“ zusammengelegt.
Der Quellcode zur Nutzung des eingesetzten Lemmatizers kann auf der beigefügten CD unter Anhang A2 nachgelesen werden.

[3]Der NEGRA Korpus basiert auf 20.602 Sätzen deutscher Zeitungstexte der Frankfurter Rundschau [28]

2.3.4 Wikigraph-Framework

Das **WikiGraph-Framework** von B. Trinczek dient der Darstellung der Wikipedia als semantisches Netz. Es bietet unter anderem die Funktionalität, Wikipedia-Konzepte als graphenspezifische Knoten und Artikel-Links als Kanten abzubilden [27].
Im Allgemeinen sind die Bezeichnungen eines Knotens innerhalb eines Graphen eindeutig. Dies hat den Vorteil, dass die Bezeichnung eines Knotens als Identifikation herangezogen werden kann. Jeder Knoten im Graphen ist typisiert und hat spezielle Eigenschaften, welche ihn von anderen Knoten im Graphen unterscheiden.

Es existieren folgende Knoten-Typen:

- ARTICLE
- EXTENDED_CONCEPT
- ICD10_CODE
- MESH_CODE
- CATEGORY
- REDIRECT

Die wichtigsten Knoten für diese Arbeit sind dabei Knoten vom Typ *ARTICLE*, welche die Artikelnamen beinhalten, und *EXTENDED_CONCEPT*, welche nach erfolgreicher Terminologie-Extraktion in den Graphen eingefügt werden.

Die im WikiGraph vorhandenen Knoten sind durch Kanten miteinander verbunden. Dabei ist es auch möglich, dass mehrere Kanten zwischen zwei Knoten existieren um zu veranschaulichen, dass auf verschiedenen Wegen zwischen den beiden Knoten navigiert werden kann. Kanten im WikiGraph sind unidirektional, d.h. die Navigation zwischen zwei Knoten ist nur in eine Richtung möglich. Des Weiteren sind alle Kanten gewichtet und können je nach Anwendungsfall unterschiedlich interpretiert werden. Aktuell sind alle Kanten standartmäßig mit dem Gewicht *DEFAULT_WEIGHT* gewichtet. Die Kanten im WikiGraph sind ebenfalls typisiert:

- ARTICLE_TO_ARTICLE
- ARTICLE_TO_TEMPLATE
- ARTICLE_TO_TEXT
- CATEGORY_TO_ARTICLE
- CATEGORY_TO_CATEGORY
- ARTICLE_TO_EXTENDED_CONCEPT

- CATEGORY_TO_EXTENDED_CONCEPT
- EXTENDED_CONCEPT_TO_EXTENDED_CONCEPT
- REDIRECT
- UNKNOWN

Der Kantentyp *ARTICLE_TO_EXTENDED_CONCEPT* stellt für diese Arbeit den wichtigsten Kantentyp dar. Nach erfolgreicher Terminologie-Extraktion werden die neu eingefügten Knoten (siehe oben) durch Kanten dieses Typs miteinander verbunden.

Es folgt eine Funktionsübersicht des WikiGraph-Frameworks [27].

- Erzeugung und Speicherung eines Wikipedia-Snapshots beliebiger Kategorie(en)
- Laden/Speichern eines Wikipedia-Snapshots
- Bearbeitung des WikiGraphen:
  a. Hinzufügen von neuen Knoten und/oder Kanten
  b. Löschen von Knoten und/oder Kanten
  c. Gewichtung von Kanten
- Überprüfung, ob Knoten oder Kanten bereits im Graph vorhanden sind

Das WikiGraph-Framework stellt einen Grundstein für die Implementierung des WikiGraph-TE-Tools dar. Alle entwickelten Algorithmen sind auf das Vorhandensein der Wikipedia in Form eines WikiGraphs abgestimmt und greifen auf Methoden dieses Frameworks zu. Auf die Nutzung verschiedener Funktionen des Frameworks wird in Kapitel 4 eingegangen.

2.3.5 Etablierte Ansätze der Terminologie-Extraktion

Bevor auf den Entwurf und die Implementierung des WikiGraph-TE-Tools eingegangen wird, erfolgt in diesem Abschnitt ein Überblick über bereits extistierende und etablierte Ansätze der Terminologie-Extraktion und somit den Stand der Forschung.
Dabei wird zwischen rein statistischen, rein linguistischen und hybriden Ansätzen unterschienden.

2.3.5.1 Statistische Ansätze

Ansätze, die auf einem rein statistischen Maß beruhen, werden meist im Information Retrieval (Abk. IR, engl. für Informationsrückgewinnung) eingesetzt und haben das Ziel, Dokumente automatisch zu indexieren, um bestehende Informationen aufzufinden [39][42].

Bookstein

Der statistische Ansatz zur automatischen Indexierung von Dokumenten von A. Bookstein und D. R. Swanson stammt aus dem Jahr 1974. Die Grundidee des Verfahrens ist, dass es zu Clustering, einer Anhäufung signifikanter Terme in wenigen Dokumenten kommt. Die Datenlage der Forschung bestand aus einer Dokumentenkollektion von hunderten wissenschaftlichen Arbeiten, welche aus unterschiedlichen Fachrichtungen zusammengesetzt wurde [8].
Die Annahme war, dass bestimmte Wörter, wie zum Beispiel *lasers* in einigen Dokumenten sehr häufig, in anderen Dokumenten jedoch überhaupt nicht auftraten. Unspezifische Wörter, wie z.B. *obtain* (engl. für „erlangen“, „erhalten“) würden nach Annahme von Bookstein relativ gleichmäßig über alle Dokumente verteilt vorkommen.
Die Brauchbarkeit eines Wortes als Indexterm für ein Dokument korreliert laut Bookstein mit der ungleichmäßigen Verteilung über alle Dokumente. So eignen sich Wörter wie *obtain* nicht als Indexterm, da durch die gleichmäßige Verteilung keine wichtigen und charakterspezifischen Informationen über den Inhalt eines Dokuments bzw. eines Artikels hervorgehen. Das Wort *lasers* hingegen eignet sich als Indexterm.
Bookstein führte den sogenannten Clusterwert x ein, ein Maß für die Anhäufung eines Wortes w in bestimmten Dokumenten:

$$x(w) = \sum_j \binom{j}{2} s_j(w) = s_2 + 3s_3 + 6s_4 + ... \tag{2.4}$$

dabei steht s_j für die Anzahl der Dokumente, in denen das Wort w exakt j-mal auftritt. Der Clusterwert x kann, laut Bookstein, als Analogie von chemischen Bindungen verstanden werden: tritt ein Wort mehrfach innerhalb eines Artikels auf, existiert eine Bindung zwischen den einzelnen Vorkommen. Diese Bindung soll belohnt werden. Bei j Vorkommen sind es $\binom{j}{2}$ Bindungen. Für den Fall, dass ein Wort w selten in einer Dokumentenkollektion auftritt, in den Artikeln jeweils aber häufig, wird der Clusterwert groß.

Bookstein nahm an, dass die Wörter über der Dokumentenkollektion Poisson-verteilt sind:

$$P(k) = \frac{e^{-\lambda}\lambda^k}{k!} = \frac{1}{k!}\lambda^k e^{-\lambda} \tag{2.5}$$

mit dem Erwartungswert $\lambda = \frac{R}{A}$, wobei A die Anzahl an Dokumenten ist, in denen ein Wort w genau R-mal vorkommt. Die Variable k stellt dabei die Anzahl an Termen dar, welche in den Dokumenten enthalten sind. Bookstein führte noch eine Funktion $g(x)$ ein. Diese gibt zu jedem Clusterwert $x(w)$ eines Wortes w aus, wie wahrscheinlich unter Annahme der Poisson-Verteilung des Wortes ist, dass dieser errechnete Wert x erreicht wird. Der von $g(x)$ errechnete Wert ist also klein, falls ein Wort stark von der Poisson-Verteilung abweicht. Diese starke Abweichungen der Verteilung bedeutet, dass ein Wort an wenigen Stellen gehäuft auftritt. Folglich können Wörter mit einem g-Wert, welcher unterhalb eines definierten Schwellenwert liegt, als gute Indexterme angesehen werden. Im Hinblick auf eine Terminologie-Extraktion würde dies anstatt auf Indexterme auf Fachterme schließen lassen.

Salton

Bei dem Ansatz von Salton, welcher im Jahr 1975 veröffentlicht wurde [21], handelt es sich ebenfalls um ein Verfahren zur automatischen Indexierung. Ähnlich wie Bookstein [8] sollen laut Salton Wörter als Indexterme bevorzugt werden, welche in wenigen Dokumenten, dafür dort aber gehäuft, vorkommen.
Salton nutzte bei seinem Ansatz die *term frequency* (Abk. TF) und die inverse document frequency (Abk. IDF). Die TF f_{ik} wird verwendet, um die Häufigkeit eines Term k in einem Dokument i zu erfassen. Die IDF_k, welches die Spezifität eines Terms k in einer Dokumentenkollektion ausdrückt, wird folgendermaßen berechnet:

$$IDF_k = log\frac{N}{d_k} + 1 \tag{2.6}$$

dabei ist N die Gesamtzahl an Dokumenten in der Dokumentenkollektion, d_k stellt die Anzahl an Dokumenten dar, welche den Term k enthalten. Falls der Term k in wenigen Dokumenten auftritt, wird IDF_k groß.
Salton verwendet das Produkt aus TF und IDF_k zur Gewichtung des Terms k in Dokument i:

$$w_{ik} = f_{ik} * IDF_k \tag{2.7}$$

Diese Gewichtung entspricht in etwa dem von Bookstein berechneten Clusterwert und wird im Gegensatz zu dessen Ansatz direkt zur Gewichtung eines Terms verwendet. Auf eine aufwendige Berechnung der Wahrscheinlichkeiten unter Annahme einer Poisson-Verteilung wird verzichtet.
Das von Salton eingesetzte tf-idf-Maß wird auch heute noch in vielen IR-Systemen eingebettet [26] .
Im Kontext dieser Arbeit würde ein großer IDF-Wert auf einen Fachterm hinweisen.

Differenzanalyse

Bei der Differenzanalyse wird gemessen, wie häufig ein Wort in einem gegebenen Textkorpus vorkommt. Dies wird mit der allgemeinsprachlichen Häufigkeit verglichen. Als Ausgangspunkt für die Differenzanalyse dienen zwei Textmengen: der Analysekorpus und der Referenzkorpus [17].
Der Analysekorpus besteht je nach Aufgabe aus Fachtexten oder, wie im Fall dieser Arbeit, aus Rohtexten von Wikipedia Artikeln, aus denen Fachterme extrahiert werden sollen.
Der Referenzkorpus hingegen ist ein allgemeinsprachlicher Textkorpus, welcher zum Beispiel aus Zeitungstexten zusammengesetzt wurde. Der hier verwendete Referenzkorpus wird, wie schon erwähnt, von der Universität Leipzig bereitgestellt.
Durch Berechnung und Vergleich von Auftrittswahrscheinlichkeiten einzelner Wortformen und Wortformkombinationen für Referenz- und Analysekorpus lassen sich vier Klassen von Wortformen bilden [17].

1. In Klasse 1 sind Wortformen, die im Referenzkorpus nicht enthalten sind. Wortformen welche dieser Klasse zugehören sind mit hoher Wahrscheinlichkeit Fachausdrücke des Themengebiets, die dem Analysekorpus zugrunde liegen.

2. Klasse 2 beinhaltet Wortformen, welche im Analysekorpus relativ gesehen häufiger vorkommen als im Referenzkörper. Wortformen dieser Klasse sind mit einer gewissen Wahrscheinlichkeit Fachterme. Um diese Wortformen identifizieren zu können, müssen ein oder mehrere Schwellwerte definiert werden, wie zum Beispiel eine Mindestdifferenz der Häufigkeitsklassen oder eine Mindesthäufigkeit einer Wortform im Analysekorpus.

3. In Klasse 3 sind Wortformen eingeordnet, welche in etwa mit der gleichen relativen Häufigkeit in beiden Textkorpora vorkommen. Dabei handelt es sich meist um Stoppwörter, d.h. Wortformen der geschlossener Wortklassen Artikel, Präpositionen oder Konjunktionen. Von Elementen dieser Klasse gehen keine themenspezifische Inhalte des Analysekorpus aus.

4. Klasse 4 beinhaltet Wortformen, welche im Analysekorpus mit geringerer Häufigkeit auftreten als im Referenzkorpus. Im Allgemeinen kann man davon ausgehen, dass diese Klasse keine Fachterme beinhaltet.

Für die Differenzanalyse sind deshalb nur Wortformen von Bedeutung, die sich in Klasse 1 und 2 befinden [17].

2.3.5.2 Linguistische Ansätze

Bourigault

Didier Bourigault veröffentlichte seinen rein linguistischen Ansatz zur Terminologie-Extraktion 1992. Er entwickelte das Softwaresystem LEXTER, welches zur Pflege von französischen Thesauri eingesetzt wurde [9].
Laut Bourigault werden Konzepte vollständig und eindeutig von terminologischen Einheiten repräsentiert. Um Mehrdeutigkeiten zu vermeiden, müssen die Termini eine bestimmte Form besitzen. Dabei handelt es sich meist um Nominalphrasen, welche aus Substantiven, Adjektiven und bestimmten Präpositionen bestehen. Andere Wortarten wie Verben, Artikel, Adverbien etc. dürfen nicht enthalten sein. Bourigault verwendet diese bestimmte Form zur Extraktion, in einem Verfahren welches in zwei Schritte unterteilt ist:

1. Analyse durch Identifikation von Grenzen: unter Verwendung eines POS-Taggers (vgl. Kapitel 2.1.6) werden im Text mit Hilfe sogenannter "frontier marker"maximale Nominalphrasen erkannt. Diese „frontier marker“ sind laut Bourigault Wörter oder Zeichen, welche mit sehr großer Sicherheit nicht in potentiellen Fachtermini auftreten, wie z.B. konjugierte Verben oder Pronomen.

2. Parsen der gefundenen maximalen Nominalphrasen mit dem Ziel der Herausfilterung von terminologischen Einheiten. Dies geschieht durch von Bourigault entwickelte Regeln, die mit Hilfe eines umfangreichen Textkorpus evaluiert wurden. Als Beispiel für solch eine Parsing-Regel führt Bourigault auf:

$$noun1\ adj\ prep\ det\ noun2\ prep\ noun3 \rightarrow \begin{cases} noun1\ adj \\ noun2\ prep\ noun3 \end{cases} \tag{2.8}$$

   Angewandt auf die maximale Nominalphrase „disque dur de la station de travail“, wird diese in ihre Bestandteile "disque dur“ und "station de travail“ zerlegt.

Arppe

Der Ansatz von Antti Arppe aus dem Jahr 1995 setzt ebenfalls wie Bourigault bei der Extraktion von maximalen Nominalphrasen aus (hier englischen) Texten an. Dazu verwendet er ein Werkzeug namens „NPtool“, welches an der Universität Helsinki entwickelt wurde [5]. Das Tool ist in der Lage, alle Teil-Nominalphrasen zu extrahieren und diese, geordnet nach deren Frequenz des „grammatikalischen Kopfs“, auszugeben. Dieser überträgt die grammatikalischen Eigenschaften auf das ihm zugrunde liegende Wort.

So wird zum Beispiel die maximale Nominalphrase

„exact form of the correct theory of quantum gravity“

durch das *NPtool* in folgende Teil-Nominalphrasen zerlegt:

exact form of the correct theory
exact form
form
form of the correct theory of quantum gravity
form of the correct theory
correct theory of quantum gravity
correct theory
theory of quantum gravity
theory
quantum gravity
gravity

Auf Grund der sehr großen Anzahl an Teil-NPs (Faktor 3 zu Anzahl maximaler NPs) ging Arppe dazu über, die POS-Muster der gefundenen Terme zu untersuchen. Gemäß seiner Angaben lassen sich ca. 80% der gefundenen Terme durch eines der folgenden POS-Muster beschreiben (dabei steht A für Adjektiv, N für Nomen und P für Präposition):

AN NN N NPN ANN

Arppe fand heraus, dass kurze POS-Muster einen hohen Recall, aber eine geringe Precision besitzen (vgl. Kapitel 5.3). Beispiele hierfür sind die POS-Muster *N* oder *A N*. Längere POS-Muster, wie zum Beispiel *NNPN* sind hingegen sehr genau, kommen aber selten vor. Berechnet man den sogenannten *F-Wert*, den Mittelwert aus Precision und Recall, zeigt sich, dass die POS-Muster *NN, AN und ANN* die am besten geeigneten Muster zur Terminologie-Extraktion darstellen.

2.3.5.3 Hybride Ansätze

Die hybriden Ansätze kombinieren statistische und linguistische Verfahren miteinander.

Daille

Der Ansatz von Beatrice Daille, Eric Gaussier und Jean-Marc Langé stammt aus dem Jahr 1994 [12]. Es handelt sich um einen der richtungsweisensten Ansätze zur Terminologie-Extraktion. Dabei wurde auf die englische und die französische Sprache eingegangen. Im Folgenden werden die Ergebnisse für das Englische erläutert.

Es wird davon ausgegangen, dass es sich bei Fachtermen um Nominalphrasen handelt, welche einer begrenzten Anzahl an syntaktischen Mustern entsprechen. Diese Muster definieren laut Daille die morphosyntaktischen Strukturen von sogenannten „multi-word-units“ (MWUs). Eine MWU besteht aus einem Hauptelement, welches ein Substantiv, ein Adjektiv, ein Verb oder ein Adverb sein kann. Daille untersucht dabei MWUs der Länge 2, welche als „*base MWUs*“ bezeichnet werden. Es handelt sich hierbei um die POS-Mustern *A N* (z.B: multiple access) und *N N* (z.B. terminology extraction). Nach der Extraktion der „*base MWUs*“ werden diese in jeweils ein geordnetes Paar, welches aus zwei Lemmata (Grundformen) besteht, überführt. Unter diesen Paaren wird das Vorkommen aller „*base MWUs*“des Korpus gespeichert. So sind zum Beispiel unter dem Paar (interference, level) folgende Sequenzen gespeichert: *interference level, interference levels, level of in- terference, levels of interference* [12]. Es konnten jedoch nicht alle durch diese Methode gefundenen Terme als relevant angesehen werden.
Aus diesem Grund war Daille auf der Suche nach einem geeigneten statistischen Maß zur Bestimmung der „*unithood*“ (engl. für Zussammengehörigkeit) der gefundenen Paare, um diese als Filter einsetzen zu können. Es wurden verschiedene statistische Maße, wie z.B. „likelihood ratio“, untersucht und verglichen. Kombinationen verschiedener Maße wurden ebenfalls analysiert.
Als Ergebnis der Untersuchungen wurde festgestellt, dass es keine Kombination verschiedener statistischer Maßzahlen gibt. Die beste Maßzahl zur Bestimmung der *unithood* ist laut Daille die Bestimmung der Häufigkeit des Auftretens des Paares im Text.
Als beste statistische Prüfgröße wurde die *likelihood ratio* genannt.

Heid

Der Ansatz von Ulrich Heid aus dem Jahr 1998 ist ein Verfahren zur Terminologie-Extraktion für die deutsche Sprache. Die Terminologie Extraktion wird in mehreren Schritten vollzogen (vgl. [16]):
Zunächst erfolgt die Extraktion von domänenspezifischen Morphemen. Darunter enthalten Basismorpheme und Affixe wie *-itis*, **trieb**, **fahr** oder *-ator*. Dazu wurde eine Differenzanalyse (vgl. Seite 28) für Einwortterme durchgeführt. Das Ergebnis ist eine erste Liste von Fachtermini. Die Komposita und Derivate unter einzelnen Fachtermen werden nun in ihre Morpheme zerlegt. Ein Beispiel dafür ist „Lungen - funktions - test“ oder „Achs - getriebe - öl“. Alle Morpheme, deren Frequenzen über einem vorher definierten Schwellenwert liegen, sind gemäß Heid domänenspezifisch.
Im Folgenden werden weitere Einwortterme aus den zu untersuchenden Texten extrahiert, welche die im voran gegangenen Schritt extrahierten Morpheme oder Affixe enthalten.
Im letzten Schritt werden Phrasen im Text extrahiert, welche bestimmten POS-Mustern folgen. Folgende POS-Muster wurden verwendet:

- N Det N^{Gen} („Anlieferung der Ware“)
- N P N („Institut für Sprachwissenschaft“)
- A N („grüner Star“)

Es werden jedoch nur Phrasen akzeptiert, welche mindestens einen Term enthalten, der in der im ersten Schritt durchgeführten Differenzanalyse als Fachterm anerkannt wurde. Heid betont, dass es insbesondere bei dem POS-Muster A N wichtig ist, dass das Adjektiv domänenspezifisch ist, da häufig auftretende Adjektive, wie z.B. „folgend“ oder „jeweilig“, nur in uninteressanten Nominalphrasen auftreten. Um dies sicherzustellen, verwendet Heid bei seiner Implementierung eine Stoppliste mit uninteressanten, häufig auftretenden Adjektiven.

2.3.5.4 Bewertung der Ansätze

Es folgt nun eine kurze Bewertung der verschiedenen Ansätze.

Wie bereits erwähnt, sind rein statistische Ansätze meist im Bereich des Information Retrievals angesiedelt. Die von Bookstein [8] und Salton [21] entwickelten Verfahren beruhen auf der Annahme, dass auf einer Dokumentenkollektion gearbeitet wird. Dies bedeutet, dass die Ansätze bzw. die darin beschriebene Formeln nicht direkt auf eine Terminologie-Extraktion mit Analysekorpus übertragen werden können, da kein einheitlicher Referenzkorpus vorhanden ist.
Die Differenzanalyse, welche von Heyer et. al. [17] beschrieben wird, wurde bereits in der vorangegangenen Studienarbeit [15] angewandt und für ein geignetes Verfahren zur Terminologie-Extraktion medizinischer Fachterme befunden.

Der rein linguistische Ansatz von Bourigault [9] verwendet eine einfache syntaktische Analyse des Textes zur Auffindung von maximalen Nominalphrasen via „frontier markers“ in Kombination mit einem POS-Tagger zur Filterung von NPs. Dieser Ansatz ist sicherlich sehr effizient, jedoch wird sehr viel Vorarbeit beim Erstellen der Regeln für den Parser benötigt. Ein Transfer vom Französischen ins Deutsche und Englische wäre also sehr aufwendig, da dort verschiedene Grammatiken gelten.
Arppe's Ansatz [5] basiert auf Verwendung eines POS-Taggers und untersucht die flache Terminologie-Extraktion mit Hilfe von POS-Mustern. Die Forschung gibt einen sehr guten Überblick darüber, welche POS-Muster für die englische Sprache verwendet werden sollten.

Der hybride Ansatz von Daille [12] gibt einen Hinweis darüber, dass einfache Methoden (Häufigkeitsquotient als nicht statistisches Maß mit den besten Ergebnissen) die besten sind.
Heid's Ansatz [16] ist ebenfalls sehr interessant, da Affixe, wie zum Beispiel *-itis* oder *-aemia* charakteristisch für medizinische Fachtermini sind.

Für diese Arbeit wurde die Differenzanalyse zur Extraktion von Fachtermini ausgewählt. Als Maß zur Bestimmung, ob ein Term relevant ist oder nicht, wird der von Daille [12] favorisierte Häufigkeitsquotient eingesetzt. Die verwendeten POS-Muster orientieren sich an Heid [16] für die deutsche Sprache und Arppe [5] für die englische Sprache.

3 Entwurf

In diesem Kapitel wird auf den Entwurf des *WikiGrap-TE-Tools* eingegangen.

Das entwickelte Programm beruht auf einem hybriden Ansatz zur Terminologie-Extraktion mit Hilfe eines POS-Taggers und Differenzanalyse via Häufigkeitsquotient. Als Referenzkorpus für die Differenzanalyse wird die von der Universität Leipzig bereitgestellte Satzkollektion verwendet, welche aus 10 Mio. Sätzen verschiedener Zeitungsartikel zusammengetragen wurde. Dieser ist sowohl für die englische, als auch für die deutsche Sprache erhältlich und kann frei bezogen werden [30].
Der Analysekorpus besteht aus Rohtexten von Wikipedia-Artikeln, welche im Bereich Gesundheit angesiedelt sind. Der Gesundheitssektor der Wikipedia umfasst in der deutschen Sprache ca. 92.000 Artikel; in der englischen Sprache sind es ungefähr 200.000 Artikel.
Nach der Termiologie-Extraktion ist die entwickelte Applikation in der Lage, einen bestehenden Gesundheitsgraphen mit den durch die Terminologie-Extraktion erhaltenen Fachterme zu erweitern.
Im Folgenden wird ein Überblick über den Prozessablauf der zu entwickelnden Applikation (Kapitel 3.1) sowie deren modularen Aufbau (Kapitel 3.2) aufgezeigt.

3.1 Prozessablauf

Der Gesamtprozess des *WikiGraph-TE-Tools* kann in drei Schritte aufgeteilt werden:
die Erzeugung des Referenz- und Analysekorpuses (Kapitel 3.1.1), die Terminologie-Extraktion und die Analyse der Ergebnisse (Kapitel 3.1.2) und die Erweiterung des Gesundheitsgraphen (Kapitel 3.1.3).
Abbildung 3.1 zeigt den vollständigen Prozessablauf der Applikation.

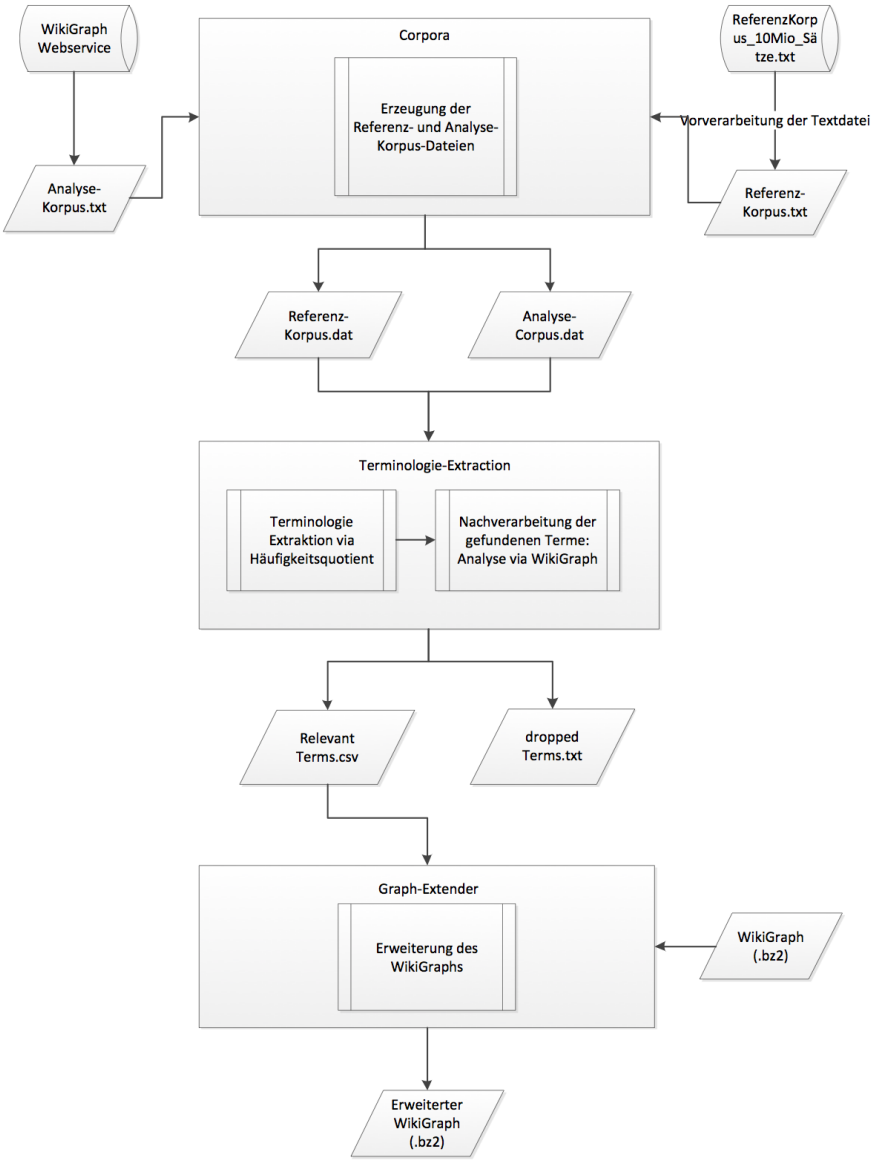


Abbildung 3.1: Vollständiger Prozessablauf des WikiGraph-TE-Tools

3.1.1 Schritt 1: Erzeugung der Korpora

In diesem Schritt wird aus den Sätzen der Leipziger Satzkollektion sowie den Rohtexten der Wikipedia-Artikel ein Referenz- und ein Analysekorpus erstellt. Bei der Erzeugung der Korpora können zwei Einstellungen vorgenommen werden:

- Verwendung des Lemmatizers zur Reduzierung der NPs auf deren lexikalische Grundform
- Ersetzung von mediznischen Abkürzungen durch deren Vollschreibweise, wie z.B. $CT \rightarrow$ *Computertomographie*

Um die Abkürzungen medizinischer Fachterme ersetzen zu können, wurde eine Liste mit ca. 1300 Abkürzungen für die deutsche Sprache, sowie für die englische Sprache eine Liste mit ca. 1400 Abkürzungen und Vollschreibweisen erarbeitet. Mehrdeutige Abkürzungen, beispielsweise die englische Abkürzung *AED* für *automated external defibrillator* oder *antiepileptic drug(s)*, wurden dabei nicht berücksichtigt, da es bei einem automatischen Verfahren nicht ohne Weiteres möglich ist, den Kontext des Artikels zu erkennen.
Der erzeugte Korpus wird als Objekt der Klasse *CorpusImpl* repräsentiert. Das korrespondierende Klassendiagramm ist in Abbildung 3.2 dargestellt.

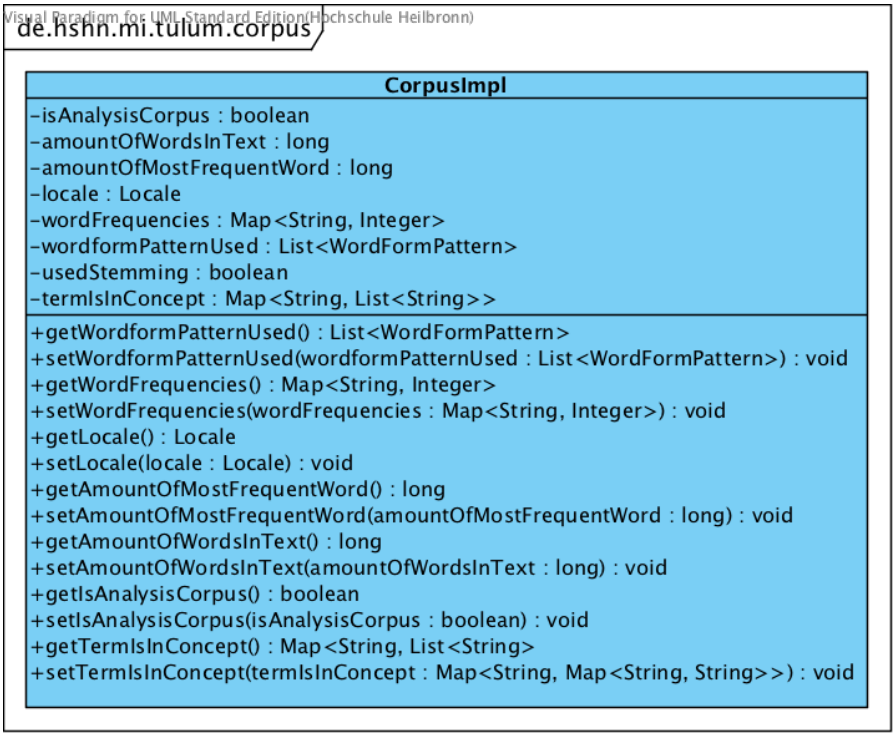


Abbildung 3.2: Klassendiagramm: CorpusImpl.java

Im Folgenden wird auf die Bedeutung der Variablen näher eingegangen:

isAnalyseCorpus	→	Spiegelt wieder, ob bei es sich es sich um einen Analysekorpus handelt
amountOfWordsInText	→	Gesamtwortzahl im Korpustext
amountOfMostFrequentWort	→	Auftrittshäufigkeit des am meisten vorkommenden Wortes der Sprache
locale	→	Sprache des Korpus
wordFrequencies	→	Ermittelte Worthäufigkeiten
wordFormPatternUsed	→	Verwendete POS-Muster
usedStemming	→	Spiegelt wieder, ob die Grundformreduzierung genutzt wurde
termIsInConcept	→	Inverted Index

Da es gewisse Unterschiede bei der Erzeugung der beiden Korpora gibt, wird auf jede gesondert eingegangen.

3.1.1.1 Erzeugung des Referenzkorpus

Die zur Erzeugung des Referenzkorpus benötigen Sätze liegen als Textdatei vor. Jeder Satz ist durch eine ID gekennzeichnet und wird gesondert in einer Zeile abgespeichert. In einem Vorbereitungsschritt werden die Satz-IDs sowie evtl. in den einzelnen Sätzten vorhandene Zeichen, wie z.B „(“ oder „]“ herausgefiltert, da sich diese Zeichen in weiteren Verarbeitungsschritten als störend herausgestellt haben. Optional besteht die Möglichkeit medizinische Abkürzungen ersetzen zu lassen. Die daraus resultierenden vorverarbeiteten Sätze werden erneut in kleineren Textdateien gespeichert, deren Dateigröße vom Benutzer festgelegt werden kann. In späteren Verarbeitungsschritten ist es daher möglich, mit kleineren Dateigrößen zu arbeiten.
Nach dem Vorverarbeitungsschritt werden die einzelnen Textdateien erneut eingelesen und in die einzelnen Sätze aufgeteilt. Die Sätze werden nun mittels POS-Tagger getaggt. Falls eine Nominalphrase vorher definierten POS-Mustern entspricht, werden diese extrahiert und optional per Lemmatizer auf ihre lexikalische Grundform reduziert. Die so erhaltenen Terme werden abgelegt. Zusätzlich wird die Information gespeichert, wie häufig der Term bereits verarbeitet wurde.
Sind alle Sätze verarbeitet, wird aus den generierten Informationen der Auftretenshäufigkeiten der extrahierten Terme, der Gesamtanzahl der Wörter in den Sätzen sowie weiteren Metainformationen (vgl. Abbildung 3.2) eine Korpusdatei erstellt.

3.1.1.2 Erzeugung des Analysekorpus

Die zur Erzeugung des Analysekorpus benötigten Rohtexte der Wikipedia-Artikel werden über den *WikiGraph-Webservice* bezogen. Dazu wird zunächst eine Liste mit allen im aktuellen Gesundheitsgraphen vorhandenen Artikelnamen angefordert. Mit Hilfe dieser Liste wird nun jeder Rohtext des korrespondierenden Artikelnamen via Webservice

bezogen. Jeder Artikel-Rohtext wird, nachdem er übertragen wurde, einer Filterung unerwünschter Satzzeichen unterzogen. Optional werden auch hier medizinische Abkürzungen durch ihre Vollschreibweise ersetzt. Die einzelnen Artikel werden mit deren Artikelnamen in Textdateien definierter Größe gespeichert.
Das Zählen der Worthäufigkeiten der extrahierten Terme erfolgt weitgehend analog zur Generierung des Referenzkorpus. Eine Ausnahme bildet das Anlegen eines sogenannten *Inverted Index*, welcher zu jedem extrahierten Term den korrespondierenden Artikelnamen (in dem dieser vorkommt) zuordnet.
Auch hier wird nach Verarbeitung aller Artikel eine Korpusdatei erstellt, welche zusätzlich den erstellten Inverted Index beinhaltet.

3.1.2 Schritt 2: Terminologie-Extraktion

Im Schritt der Terminologie-Extraktion werden die zuvor erstellten Korpusdateien eingelesen. Die Terme, die während der Analysekorpuserzeugung extrahiert wurden, sind absteigend nach deren Häufigkeit angeordnet. Im Folgenden wird für jeden Term, dessen Häufigkeit größer eines vorher definierten Schwellenwerts ist, eine Signifikanz durch den sogenannten Häufigkeitsquotienten berechnet [18]:

$$HQ(W) = \frac{l_w}{N} * \frac{M}{p_w} \tag{3.1}$$

dabei ist l_w die Häufigkeit des Terms w im Analysekorpus, N die Gesamtanzahl an Wörtern im Analysekorpus, M die Gesamtzahl an Wörter im Referenzkorpus und p_w die Häufigkeit des Terms im Referenzkorpus. Der Häufigkeitsquotient $HQ(w)$ wird also groß, falls ein Wort häufig im Analysekorpus, aber nur selten im Referenzkorpus vorhanden ist. Es handelt sich hierbei um keine statistische Prüfgröße, jedoch haben die Forschungen von Daille („The simple frequency count of the pair turns out to be the best score.“) [12] , Damerau („...a simple ratio of subject matter relative frequency to total sample relative frequency is about as good as more elaborate calculations, and in some instances superior.“) [13] und Witschel („Der HQ liefert allerdings für den medizinischen Text bessere Ergebnisse als die LR.“) [40] gezeigt, dass bei der Extraktion medizinischer Fachterme der Häufigkeitsquotient die besten Ergebnisse liefert.
Alle Terme, deren Signifikanz oberhalb eines definierten Schwellwerts liegen, werden als Fachterme anerkannt.
Nachdem die Fachterme durch den Schritt der Terminologie-Extraktion ermittelt wurden, werden diese einer Analyse unterzogen.
Ein Fachterm wird nur als relevant anerkannt, wenn im Gesundheitsgraph noch kein Artikelnamen vorhanden ist, welcher mit diesem Fachterm identisch ist. Ist ein Fachterm bereits als Konzept im Gesundheitsgraphen vorhanden, wird dieser in eine Textdatei namens *droppedTerms.txt* geschrieben.

Ist der gefundene Fachterm bisher nicht als Konzept im Graph vorhanden, wird der Fachterm mit den Artikelnamen, in denen er vorkommt, in die csv-Datei *relevantTerms.csv*

geschrieben. Der erste Eintrag einer Zeile steht für den gefundenen Fachterm, alle nachfolgenden, durch Semikolon getrennten Einträge, stehen für die Artikelnamen, in deren Rohtext der gefundene Fachterm vorkommt (vgl. Abbildung 5.1, S. 63). Hier kommt der in Schritt 1 erstellte Inverted Index (vgl. Kapitel 3.1.1.2) zum Einsatz.

3.1.3 Schritt 3: Erweiterung des Gesundheitsgraphen

Im letzten Schritt wird der Gesundheitsgraph erweitert. Der Gesundheitsgraph, welcher im Dateiformat *graphml.bz2* vorliegt, wird in das Programm geladen und mit Hilfe der in Schritt 3 erstellten *relevantTerms.csv* erweitert:
Der gefundene Fachterm wird als Kante vom Kantentyp *EXTENDED_CONCEPT* eingefügt und durch Kanten vom Kantentyp *ARTICLE_TO_EXTENDED_CONCEPT* und dem Gewicht *DEFAULT_WEIGHT* mit den Artikeln, in denen der Fachterm vorkommt, verbunden.
Nach erfolgreicher Erweiterung wird der erweiterte Gesundheitsgraph im Format *graphml.bz2* gespeichert.

3.2 Modulüberblick

Die Implementierung des *WikiGraph-TE-Tools* wurde mit Hilfe des Build-Tools *Maven* (vgl. Kapitel2.3.1 realisiert und folgt dem „Prinzip der Modulisierung“. Es wurden interne und externe Module zur Realisierung der Anwendung verwendet, welche in den folgenden Abschnitten erläutert werden. Abbildung 3.3 stellt eine Übersicht der verwendeten Module dar.

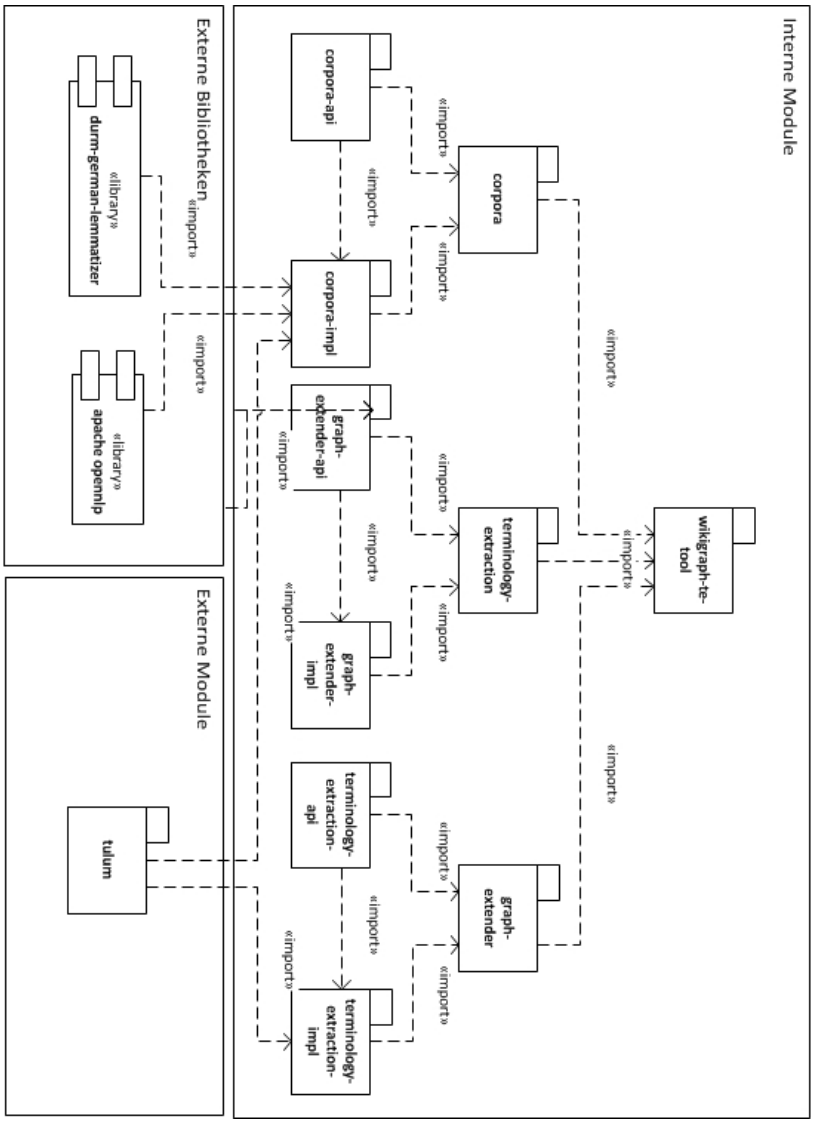


Abbildung 3.3: Modul-Übersicht des WikiGraph-TE-Tools

3.2.1 Selbsterstellte Module

Das *WikiGraph-TE-Tool* besteht aus drei unabhängigen Maven-Projekten. Jedes Projekt ist in zwei Module unterteilt: ein Modul, welches die Schnittstellen enthält und ein Modul, welches die Implementierungen der in den Schnittstellen definierten Methoden beinhaltet.

corpora-api:	Das Modul *corpora-api* definiert zentrale Schnittstellen zur Erstellung des für die Terminologie-Extraktion benötigten Referenz- und Analysekorpus. Es sind unter anderem die Funktionalitäten der Vorverarbeitung des Referenzkorpus, das Bereitstellen der Wikipedia-Rohtexte und die Erzeugung der Korpora enthalten.
corpora-impl:	Das Modul *corpora-impl* implementiert die soeben vorgestellte Schnittstelle und beinhaltet die implementierenden Klassen. Hier werden die externen Bibliotheken *durm-german-lemmatizer* und *apache-opennlp* verwendet.
corpora:	Das Projekt *corpora* verbindet die Module *corpora-impl* und *corpora-impl* zu einer funktionellen Einheit.
terminology-extraction-api:	Das Modul *terminology-extraction-api* definiert die Schnittstellen zum Zugriff auf die Funktionen zur Durchführung der Terminologie-Extraktion.
terminoligy-extraction-impl:	Das Modul *terminoligy-extraction-impl* implementiert die gerade vorgestellten Interfaces und beinhaltet die implementierenden Klassen.
terminology-extraction:	Das Projekt *terminology-extraction* wird dazu verwendet, die beiden Submodule zu vereinigen und somit ein funktionstüchtiges, eigenständiges Maven-Projekt zu erhalten.
graph-extender-api:	Das Modul *graph-extender-api* definiert zentrale Schnittstellen zum Zugriff auf die Funktion zur Erweiterung des WikiGraphen mit Hilfe der Ergebnisse der Terminologie-Extraktion. Die Interfaces werden vom Modul *graph-extender-impl* implementiert.
graph-extender:	Das Projekt *graph-extender* wiederum dient der Erstellung eines lauffähigen Maven Projekts.
wikigraph-te-tool:	Das Projekt *wikigraph-te-tool* wird dazu verwendet, die drei Teilprojekte in ein Gesamtprojekt zu vereinigen. Die Anwendung kann per Kommandozeile gesteuert und per Property-File konfiguriert werden.

3.2.2 Externe Module und Bibliotheken

Das externe Modul *tulum* steht in diesem Kontext für ein System, welches im Gesamtprojekt TULUM erarbeitet wurde. Es beinhaltet das *WikiGraph-Framework* (vgl. Kapitel 2.3.4) und den zur Beschaffung der Wikipedia-Rohtexte genutzten *WikiGraph-WS*.

Die externe Bibliothek *apache-opennlp* beinhaltet verschiedene Werkzeuge, die im Bereich der Computerlinguistik anzusiedeln sind (vgl. Kapitel 2.3.2).

Die externe Bibliothek *durm-german-lemmatizer* wird im Schritt der Korpusgeneration zur Reduzierung der Terme auf deren lexikalische Grundform verwendet (vgl 2.3.3).

4 Entwicklung und Implementierung

Dieses Kapitel beschäfftigt sich mit der Entwicklung und Implementierung des *WikiGraph-TE-Tool*.
Das Programm *WikiGraph-TE-Tool* wurde komplett in Eclipse und mit Hilfe von Java entwickelt; die vorhandenen Tests mit JUnit4 realisiert. Die Erstellung der Programmstruktur erfolgte mit Hilfe von Maven erstellt und wurde per pom.xml beschrieben.

Die Anwendung wird dazu verwendet, größe Textmengen zu prozessieren. Besonders wichtig war das Schreiben einer möglichst effizienten und performanten Implementierung. Um den Speicherverbrauch so gering wie möglich zu halten, kamen in begründeten Fällen *StringBuilder* anstatt normaler *String-Objekte* zum Einsatz. Der Vorteil des *StringBuilders* ist die effizientere Implementierung, zum Beispiel werden bei der Konkartination von Zeichenketten keine neuen Objekte angelegt [7].

Im Folgenden wird auf wichtige Aspekte der Anwendung sowie für den Prozessablauf relevante Implementierungen genauer eingegangen.

4.1 Modul: corpora-impl

Im Modul *corpora-impl* sind Funktionalitäten enthalten, welche zur Erzeugung des Referenz- und Analysekorpus benötigt werden.

4.1.1 Beschaffung der Rohtexte

In Abbildung 4.1 ist ein Codeausschnitt der Klasse *PreprocessorImpl.java* abgebildet, welcher die Beschaffung der Artikel-Rohtexte der Wikipedia aufzeigt.

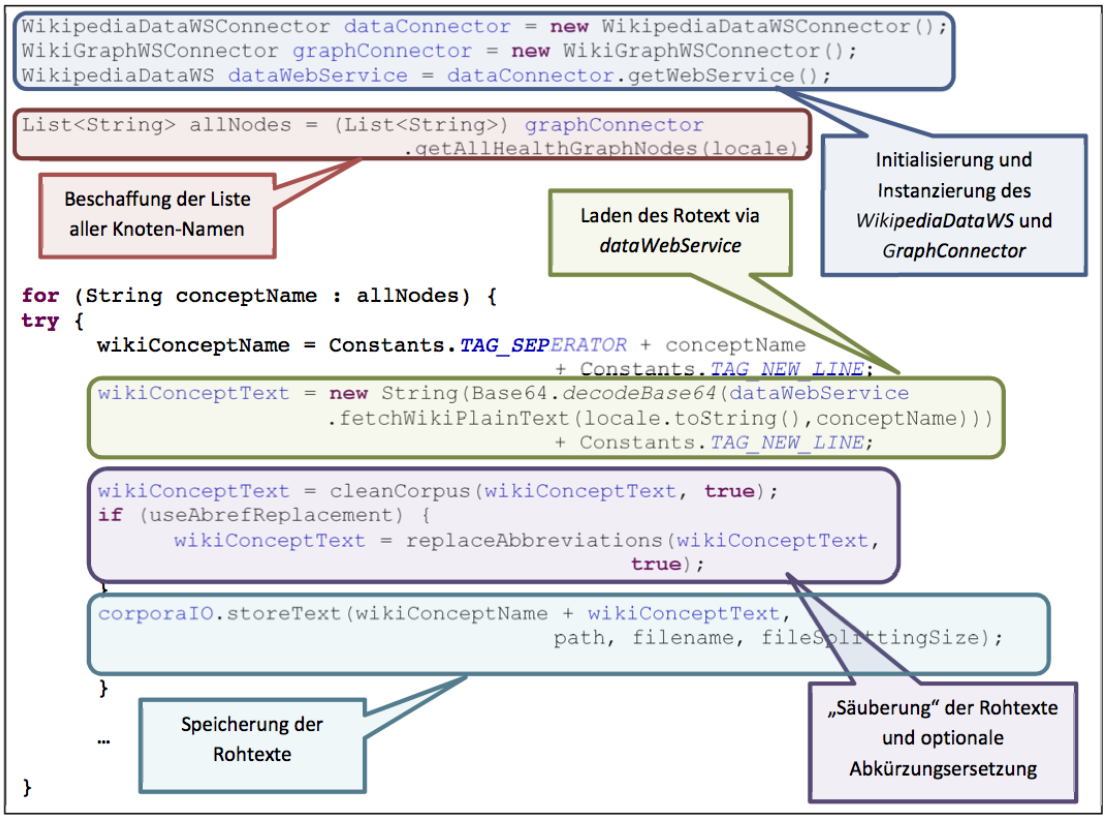


Abbildung 4.1: Codeausschnitt PreprocessorImpl.java: Beschaffung der Rohtexte

Mit Hilfe der Variable *graphConnector* kann auf den *WikiGraphWSConnector* zugegriffen werden. Durch den Methodenaufruf *getAllHealthGraphNoces(locale)* werden alle Knoten in der gewünschten Sprache geliefert und unter der Variable *allNodes* abgelegt. Die *WikipediaDataWS*-Instanz liefert im nachfolgenden Schritt für jeden Knoten den korrespondierenden Rohtext. Vor der Speicherung der einzelnen Rohtexte in einer Textdatei, werden in der Methode *cleanText()* ungewünschte Satzzeichen herausgefiltert und optional, durch den Aufruf von *replaceAbbreviations()*, medizinische Abkürzungen durch deren Vollschreibweise ersetzt.

4.1.2 Erzeugung der Korpora

Um eine Differenzanalyse durchführen zu können, wird sowohl ein Referenz- als auch ein Analysekorpus benötigt (vgl. Seite 28, Differenzanalyse).

Diese erzeugten Korpora werden durch die Klasse *CorpusImpl.java* repräsentiert (vgl. Abbildung 3.2) Neben den Worthäufigkeiten werden Meta-Informationen wie z.B. die Sprache des Korpus, die bei der Erstellung des Korpus genutzten POS-Muster, die Gesamtanzahl der in der Ressourcen-Textdatei (Wikipedia-Rohtexte bzw. Satzkollektion) enthaltenen Wörter und die Information, ob es sich bei dem Korpus um einen Referenz- bzw. Analysekorpus handelt, abgelegt.

WordFormPattern.java

Die Enumeration *WordFormPattern.java* repräsentiert die aktuell verwendbaren POS-Muster:

- NOUN_NOUN
- PERSONAL_NAME
- ADJ_NOUN
- ADJ_NOUN_NOUN
- NOUN_NOUN_NOUN
- NOUN_PREP_NOUN

TaggedWordImpl.java

Ein durch den POS-Tagger getaggtes Wort wird durch die Klasse *TaggedWordImpl.java* repräsentiert. In einem Objekt dieser Klasse ist die String-Repräsentation des getaggten Worts und die ihm zugewiesene Wortform beinhaltet.

Die Klasse *CorporaGeneratorImpl* leitet die Generierung der Korpora ein, wobei zwischen der Generierung eines Referenz- und eines Analysekorpus unterschieden wird. In beiden Fällen werden die einzelnen Textdateien, welche die Korpustexte enthalten in den Speicher geladen und weiterverarbeitet. Beim Referenzkorpus erfolgt vor dem Zählen der Worthäufigkeiten das Anlegen einer Liste welche die einzelnen Sätze des Referenzkorpustexts enthält (vgl. Abbildung 4.2). Auf Grund der Speicherstruktur der Sätze (jeder Satz ist in einer eigenen Zeile abgespeichert) kann die Zerlegung der Sätze durch Aufruf der *split()*-Methode der Klasse *String* erfolgen.

```
List<File> filesToLoad = null;
…
//Checking which files need to be loaded
filesToLoad =
corporaIO.getFilenames(referenceCorpusFilePath,referenceTextFileName);
…
String referenceCorpusText = "";
String[] referenceCorpusTextSplitted;
List<String> referenceCorpusTextList = new ArrayList<String>();

for(File file : filesToLoad){
  referenceCorpusText = corporaIO.
                loadSingleTextFile(file.getPath(),textIsCompressed);
//Splitting the text into lines
   referenceCorpusTextSplitted =referenceCorpusText.
                                    split(Constants.TAG_NEW_LINE);
   referenceCorpusSentences.addAll(
                              Arrays.asList(referenceCorpusTextSplitted);

}
```

Abbildung 4.2: Codeausschnitt CorporaGeneratorImpl: Vorbereitung zur Referenzkorpus- Erzeugung

Im Fall der Analysekorpuserzeugung wird nach dem Einlesen jeder Textdatei eine HashMap erweitert, welche die Artikelnamen der Wikipediatext auf deren Artikeltexte abbildet (vgl. Abbildung 4.3).

```
String analysisCorpusText = "";
List<File> filesToLoad = null;
…
//Checking which files need to be loaded
filesToLoad = corporaIO.
      getFilenames(analysisCorpusTextFilePath, analysisTextfileName);
…
//loading each textfile to generate a Map<ArticleName,ArticleText>
for(File file : filesToLoad){
      analysisCorpusText = corporaIO.loadSingleTextFile(file.getPath(),
                                                textIsCompressed);
      conceptMap = generateMap(conceptMap, analysisCorpusText);
}
```

Abbildung 4.3: Codeausschnitt CorporaGeneratorImpl: Vorbereitung zur Analysezkorpus- Erzeugung

Das Zählen der Worthäufigkeiten wird durch die Klassen *AnalysisCorpusWordFormFrequencyCalculatorImpl.java* und *ReferenceCorpusWordFormFrequencyCalculatorImpl.java* realisiert. Um möglichst hohe Performanz zu erlangen wurde das Zählen der Häufigkeiten parallelisiert. Abbildung 4.4 stellt die Realisierung der Parallelisierung dar.

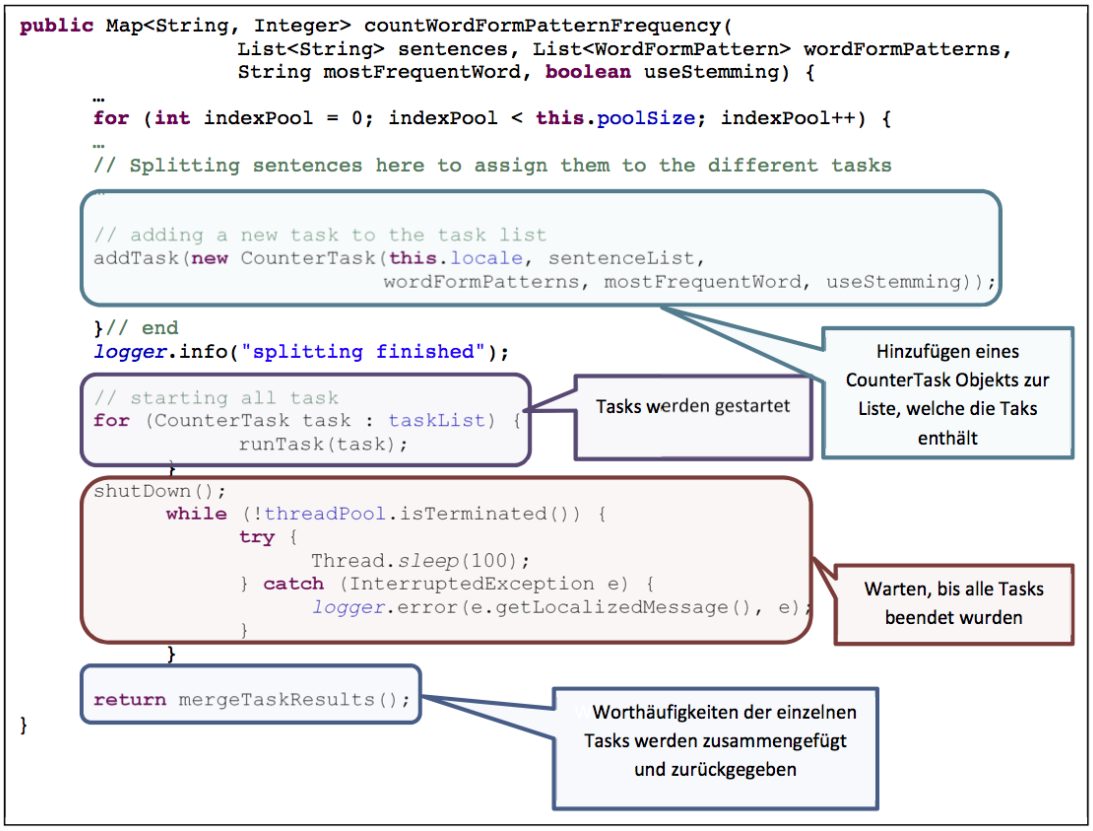


Abbildung 4.4: Codeausschnitt: Parallelisierung zum Zählen der Worthäufigkeiten

Im Fall des Referenzkorpus wird die zuvor erstellte Liste, welche die einzelnen Sätze des Referenzkorpustextes enthält, auf die einzelnen Tasks aufgeteilt. Bei der Analysekorpuserzeugung wird die zuvor erzeugte HashMap auf die Tasks aufgesplittet, welche zur parallelen Berechnung eingesetzt werden. Die vorhandene Anzahl an Prozessorkernen bestimmt die Anzahl der Tasks und legt diese in der Formel „Anzahl Rechenkerne +1“ fest. In den Tasks werden die Auftretenshäufigkeiten der Terme gezählt und nach deren Beendigung die Einzelergebnisse zusammengefügt.
Der Alogorithmus zur Häufigkeitszählung wird im Folgenden per Pseudocode aufgezeigt, da eine Darstellung mit Quellcode zu unübersichtlich wäre. Dabei stellt die Variable *wordFormPatterns* die Liste der eingesetzten POS-Muster dar. Die Ergebnismenge stellt eine HashMap dar, welche den Term als Schlüssel und die Anzahl der Häufigkeiten als Wert abbildet.

Algorithm 1 Grundalgorithmus: Zählen der Häufigkeiten

```
1: List< TaggedWord > getaggteWörter;
2: List< WordFormPatter > aufgeteiltePattern;
3: for all Sätze do
4:   getaggteWörter = von POS-Tagger getaggter Satz;
5:   for all wfp : wordFormPatterns do
6:     aufgeteiltePattern = wfp zerlegt in EinzelPatterns
7:     for int i; i < Länge getaggteWörter; i++ do
8:       if Länge wfp == 1 then
9:         if getaggteWörter[i] entspricht aufgeteiltePattern[i] then
10:          Grundformreduktion und Aufnahme in Ergebnismenge von getaggteWörter[i]
11:        end if
12:      end if
13:      if Länge wfp == 2 then
14:        if getaggteWörter[i] entspricht aufgeteiltePattern[i] and getaggteWörter[i+1] entspricht aufgeteiltePattern[i+1] then
15:          Grundformreduktion und Aufnahme in Ergebnismenge von getaggteWörter[i] + getaggteWörter[i+1]
16:        end if
17:      end if
18:      if Länge wfp == 3 then
19:        if getaggteWörter[i] entspricht aufgeteiltePattern[i] and getaggteWörter[i+1] entspricht aufgeteiltePattern[i+1] and getaggteWörter[i+2] entspricht aufgeteiltePattern[i+2] then
20:          Grundformreduktion und Aufnahme in Ergebnismenge von getaggteWörter[i] + getaggteWörter[i+1] + getaggteWörter[i+2]
21:        end if
22:      end if
23:    end for
24:  end for
25: end for
```

Bei der Erzeugung eines Analysekorpus wird, bevor über die Liste von Sätzen iteriert wird, eine umgebenden Schleife der Wikipedia-Artikel durchlaufen. Die Rohtexte werden dort mit Hilfe des *SentenceDetecors* (vgl. Kapitel 2.3.2.1) in die einzelnen Sätze aufgeteilt.

Die Ausführung der Grundformreduktion und Aufnahme in die Ergebnismenge erfolgt durch den nachfolgenden Code.

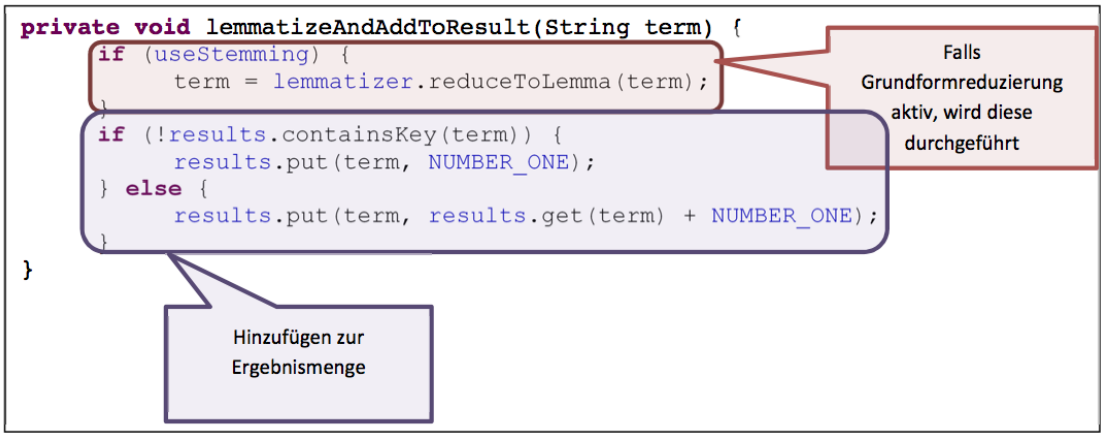


Abbildung 4.5: Codeausschnitt: Grundformreduktion und Aufnahme in Ergebnismenge

Falls die Grundformreduzierung aktiviert ist, wird der übergebene Term auf die Grundform reduziert. Ist der Term bereits in der Ergebnismenge vorhanden, wird die Häufigkeit inkrementiert; ansonsten wird ein neuer Eintrag mit dem Term als Schlüssel und der Häufigkeit 1 als Wert erzeugt.
Der in Kapitel 3.1.1.2 angesprochene Unterschied bei der Erzeugung des Analysekorpus besteht im Anlegen eines sogenannten *Inverted Index*, welcher in nachfolgendem Codeausschnitt als Variable *termIsInConcepts* deklariert ist.

```
private void addConceptToTerm(String foundTerm, String concept) {

if (this.termIsInConcepts.containsKey(foundTerm)) {
      if (!this.termIsInConcepts.get(foundTerm).contains(concept)) {
            this.termIsInConcepts.get(foundTerm).add(concept);
            }
      } else {
            List<String> list = new ArrayList<String>();
            list.add(concept);
            this.termIsInConcepts.put(foundTerm, list);
      }
}
```

Abbildung 4.6: Codeausschnitt: Erzeugung Inverted Index

Dabei werden jedem gefundenen Term die Wikipedia-Artikel zugeordnet, in denen der Term auftritt.

Nach dem Zählen der Worthäufigkeiten wird ein *CorpusImpl*-Objekt erzeugt und serialisiert gespeichert.

```
Map<String, Integer> result = analysisWordFormPatternFreqCalc
                        .countWordFormPatternFrequency(analysisCorpusText,
                                wordFormPatterns, useStemming);
//sorting word forms descending by frequency
Map<String, Integer> sortedResult = MapUtil.sortByValue(result, false);
// create corpus
Corpus analysisCorpus = new CorpusImpl();
analysisCorpus.setIsAnalysisCorpus(true);
analysisCorpus.setWordFormPatternsUsed(wordFormPatterns);
analysisCorpus.setLocale(locale);
analysisCorpus.setAmountOfWordsInText(analysisWordFormPatternFreqCalc
                                        .getTotalAmountOfWords());
analysisCorpus.setWordFrequencies(sortedResult);
analysisCorpus.setUsedStemming(useStemming);
analysisCorpus.setTermIsInConcepts(analysisWordFormPatternFreqCalc
                                        .mergeTermIsInConceptMap());
// Corpus information displayed
analysisCorpus.provideCorpusInformation();
//storing corpus on hdd
corporaIO
      .storeCorpus(analysisCorpusFilePath,analysisCorpusFileName,analysisCorpus);
```

Abbildung 4.7: Codeausschnitt CorporaGeneratorImpl.java: Erzeugung und Speicherung einer Korpusdatei

4.2 Modul: terminology-extraction-impl

Das Modul *terminology-extraction-impl* bietet Funktionalitäten zur Terminologie-Extraktion. Dies beinhaltet die Differenzanalyse sowie die angeschlossene Analyse der gefundenen Fachterme.

4.2.1 Differenzanalyse

Um die Differenzanalyse durchführen zu können, werden die Worthäufigkeiten der im Schritt der Korpus-Erzeugung gefundenen Terme benötigt. Aus diesem Grund werden die erzeugten Korpora-Dateien, wie in Abbildung 4.8 dargestellt, in das Programm geladen.

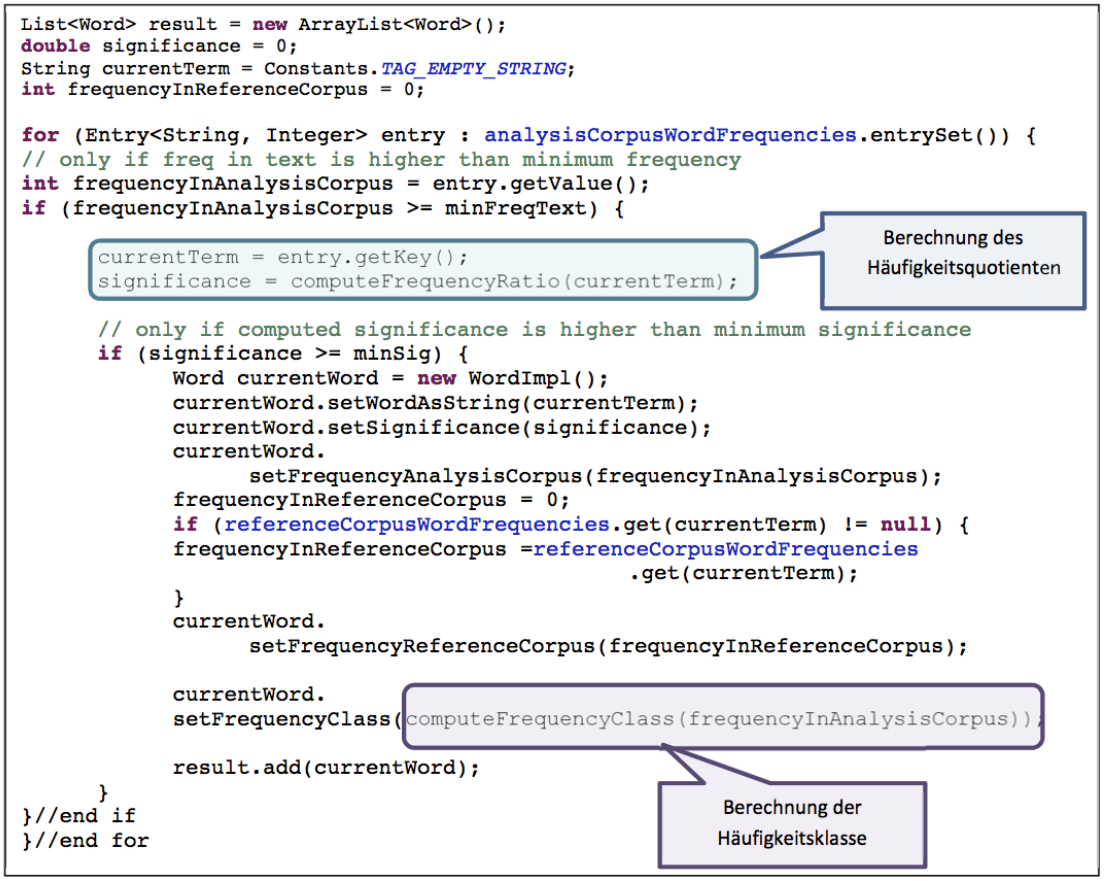
```
// loading reference corpus
Corpus referenceCorpus = corporaIO.loadCorpus(path, locale.toString()
                        + referenceCorpusFileName);
referenceCorpus.provideCorpusInformation();

// loading analysis corpus
Corpus analysisCorpus = corporaIO.loadCorpus(path, locale.toString()
                        + analysisCorpusFileName);
analysisCorpus.provideCorpusInformation();

// differential analysis
List<Word> significantTerms = diffAnalysis.performDifferentialAnalysis(
                              referenceCorpus, analysisCorpus, minSig,
                              minFreqText);
// storing results
storeDAResults(significantTerms);
```

Abbildung 4.8: Codeausschnitt TerminologyExtractionImpl.java: Methodenaufruf zur Differenzanalyse

Während der Differenzanalyse wird neben dem Häufigkeitsquotienten (vgl. Formel 3.1) auch die Häufigkeitsklassse (vgl Kapitel 2.1.5) des Terms berechnet. Alle Terme, welche die Kriterien der Schwellwerte „Mindesthäufigkeit im Analysekorpus“ (welche durch die Variable *minFreqText* dargestellt ist) und der berechneten Mindestsignifikanz via Häufigkeitsquotient (dargestellt durch die Variable *minSig*), überschreiten, sind als Fachterm anerkannt.




```
List<Word> result = new ArrayList<Word>();
double significance = 0;
String currentTerm = Constants.TAG_EMPTY_STRING;
int frequencyInReferenceCorpus = 0;

for (Entry<String, Integer> entry : analysisCorpusWordFrequencies.entrySet()) {
// only if freq in text is higher than minimum frequency
int frequencyInAnalysisCorpus = entry.getValue();
if (frequencyInAnalysisCorpus >= minFreqText) {

      currentTerm = entry.getKey();
      significance = computeFrequencyRatio(currentTerm);

      // only if computed significance is higher than minimum significance
      if (significance >= minSig) {
            Word currentWord = new WordImpl();
            currentWord.setWordAsString(currentTerm);
            currentWord.setSignificance(significance);
            currentWord.
                  setFrequencyAnalysisCorpus(frequencyInAnalysisCorpus);
            frequencyInReferenceCorpus = 0;
            if (referenceCorpusWordFrequencies.get(currentTerm) != null) {
            frequencyInReferenceCorpus =referenceCorpusWordFrequencies
                                                .get(currentTerm);
            }
            currentWord.
                  setFrequencyReferenceCorpus(frequencyInReferenceCorpus);

            currentWord.
            setFrequencyClass(computeFrequencyClass(frequencyInAnalysisCorpus));

            result.add(currentWord);
      }
}//end if
}//end for
```

Abbildung 4.9: Codeausschnitt DifferentialAnalysisImpl.java: Berechnung des Häufigkeitsquotienten und der Häufigkeitsklasse

Das Ergebnis ist eine nach Signifikanz absteigend sortierte Liste mit Objekten der Klasse *Word*. Objekte dieser Klasse beinhalten eine String-Repräsentation des Terms, die errechnete Signifikanz per Häufigkeitsquotient, die Worthäufigkeitsklasse sowie die Auftretenshäufigkeit im Referenz- und Analysetext.

4.2.2 Analyse der Fachterme

Der nachfolgende Schritt beinhaltet die Analyse aller Terme, welche sich als Fachterme qualifiziert haben. Ziel dieser Analyse ist die Vorbereitung zur Erweiterung des Gesundheitsgraphen.
Jeder Fachterm wird dabei einer Überprüfung unterzogen, ob dieser bereits als Knoten im Graph vorhanden ist. Dabei kommt eine Binärsuche zum Einsatz, bei der ein Abgleich der im Gesundheitsraph bereits vorhandenen Knotennamen mit den erhaltenen Fachtermen stattfindet. Ist der Fachterm bereits im Graphen vorhanden, erfolgt dessen Aufnahme in die Liste *droppedTermsList*, welche bereits vorhandene Fachterme beinhaltet.
Ist der aktuelle Fachterm noch nicht im Graphen vorhanden, wird dieser mit den Artikelnamen, in denen er auftritt, gespeichert und der Liste *relevantTermsList* hinzugefügt. Nachdem jeder Term diese Analyse durchlaufen hat, werden die Ergebnisse UTF-8 kodiert auf die Festplatte geschrieben. Die Speicherung aller Terme, welche in der *droppedTermsList* enthalten sind und somit als unrelevant eingestuften wurden, erfolgt in der Datei *droppedTerms.txt*. Die Liste der relevanten Terme wird in die Datei *relevantTerms.csv* geschrieben. Diese Datei ist folgendermaßen aufgebaut: jede Zeile steht für einen gefundenen Fachterm inkl. den Titel des Artikels, in denen dieser vorkommt. Der erste Eintrag der Zeile repräsentiert den gefundenen Fachterm, die darauf folgenden Terme, welche durch Semikolon getrennt sind, stehen für die Artikelnamen. Im Folgenden ist ein Eintrag schematisch aufgezeigt:

Fachterm; Artikelname1; Artikelname2; Artikelname3; Artikelname4;...

Der dazugehörige Codeausschnitt ist in Abbildung 4.10 dargestellt.

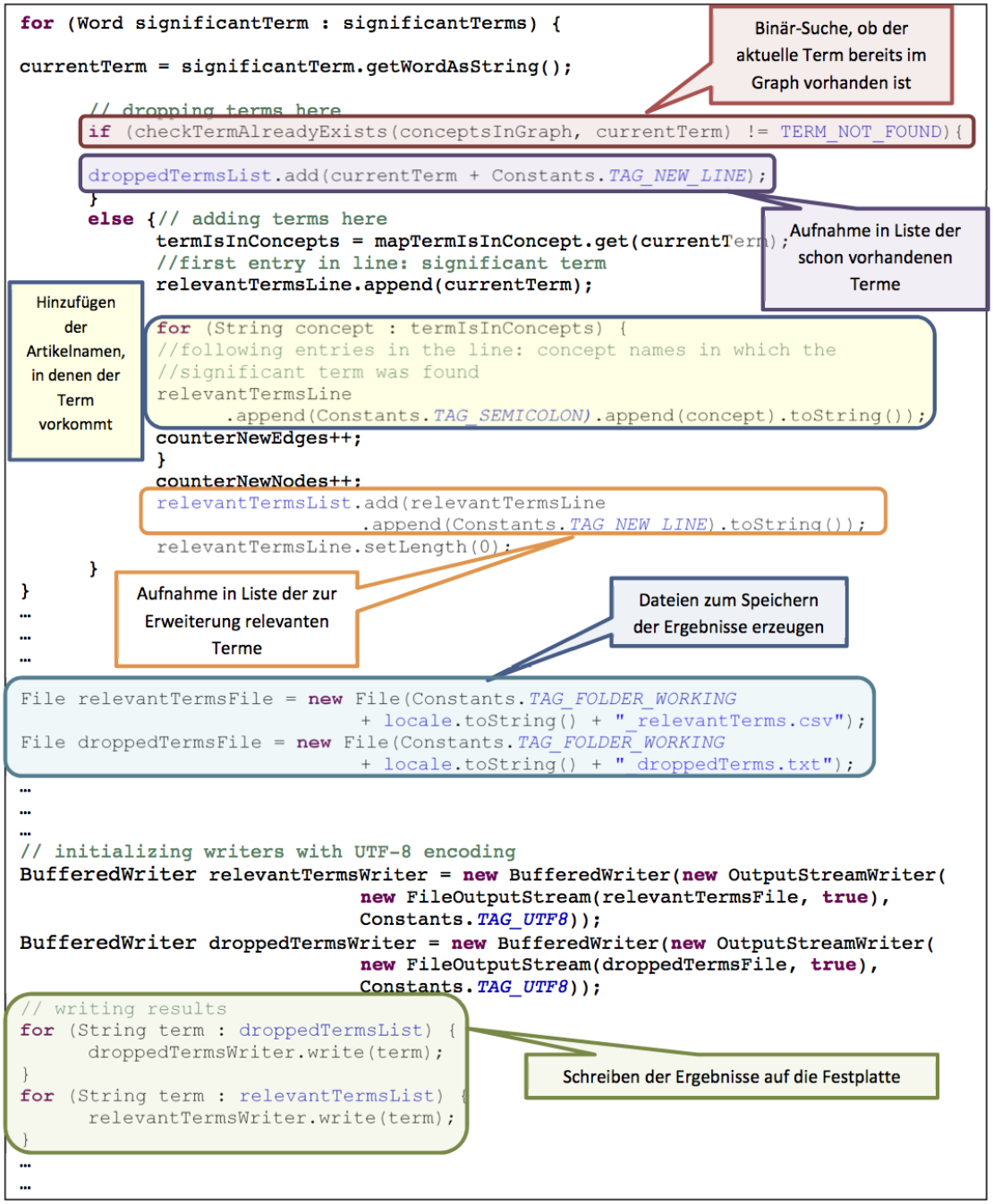


Abbildung 4.10: Codeausschnitt PostprocessorImpl.java: Analyse der gefundenen Fachterme

4.3 Modul: graph-extender-impl

Das Modul *graph-extender-impl* beinhaltet die Funktionalität zur Erweiterung des Gesundheitsgraphen und dessen anschließende Speicherung.

4.3.1 Erweiterung des Gesundheitsgraphen

Zur Erweiterung des Graphen wird die zuvor erstellte Datei *relevantTerms.csv* benötigt. Diese umfasst die Informationen, welche neuen Knoten eingefügt werden und zu welchen bestehenden Knoten Kanten gezogen werden müssen. Der zur Erweiterung benötigte *WikiGraph* wird bereits als Maven-Dependency in deutscher, sowie auch englischer Sprache mitgeliefert.

In Abbildung 4.11 ist ein Ausschnitt der Klasse *WikiGraphExtenderImpl.java* dargestellt, welche die Erweiterung des Graphen realisiert. Zunächst wird der zu erweiternde WikiGraph in das Programm geladen und der Variable *wikigraphResult* zugewiesen. Dies geschieht mit Hilfe einer Instanz der Klasse *WikiGraphSearchService*. Danach wird die Datei *relevantTerms.csv* zeilenweise in das Programm geladen und in einzelne Einträge aufgesplittet. Die darin enthaltenen Fachterme werden als neue Knoten vom Typ *EXTENDED_CONCEPT* dem WikiGraphen hinzugefügt. Danach werden neue Kanten vom Typ ARTICLE_TO_EXTENDED_CONCEPT und dem Gewicht *DEFAULT_WEIGHT* erzeugt und zwischen dem neu erzeugten Knoten und den bereits existierenden Knoten, in deren Artikel der Fachterm vorkommt, eingefügt.
Nach der Verarbeitung alle Einträge wird der erweiterte WikiGraph im Ordner *result* abgespeichert.

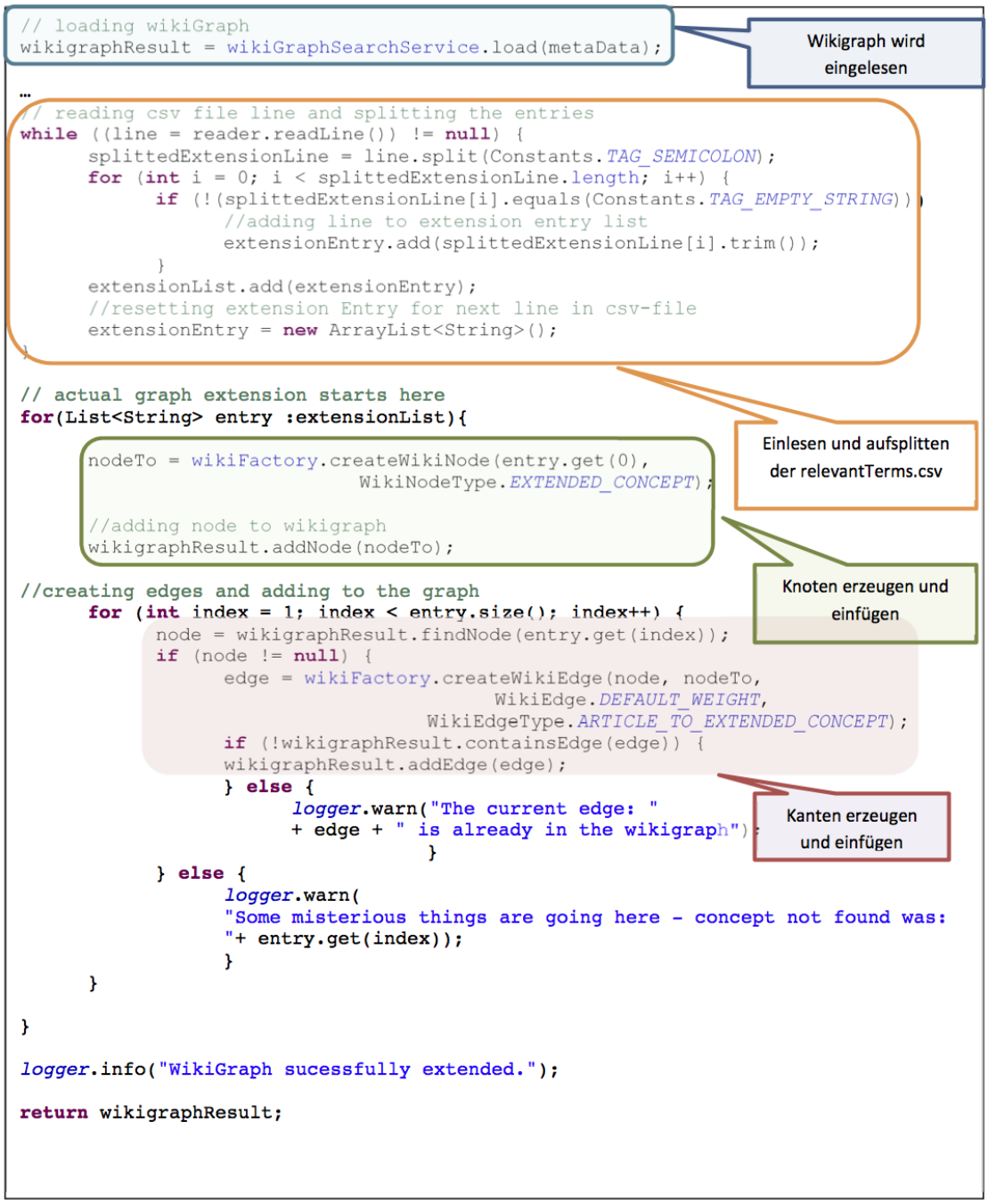


Abbildung 4.11: Codeausschnitt WikiGraphExtenderImpl.java: Erweiterung des Graphen

4.4 Verwendung des WikiGraph-TE-Tools

Das WikiGraph-TE-Tool ist eine Konsolenanwendung und muss daher per Terminalbefehl gestartet werden. Hierzu bedarf es in das Programmverzeichnis zu wechseln, welche die ausführbare jar-Datei sowie deren benötigten Ressourcen beinhaltet.

a) Fetch Wikipedia Text

Ziel dieser Funktion ist es, die Rohtexte der gesamten Gesundheitsdomäne des WikiGraphen vom Webservice zu beziehen und diese in einer Textdatei zur weiteren Verarbeitung zu speichern.
Die Funktion wird per Kommandozeilenparameter „–fwt <Anzahl Artikel>“ beziehungsweise „-fetch wikipedia text <Anzahl Artikel> aufgerufen. Der Parameter <Anzahl Artikel> ist die Anzahl der Artikel, deren Rohtexte via WebService geladen und in eine Textdatei abgespeichert werden sollen. Bei Eingabe des Befehls „-fwt all“ werden alle vorhandenen Rohtexte geladen.

b) Preprocess Reference Corpus text

Ziel dieser Funktion ist es den Referenzkorpus Text für die Weiterverarbeitung vorzubereiten, d.h. die enthaltenen Zeilennummern sowie unerwünschte Satzzeichen zu entfernen und den Text in Dateien kleinerer Größe aufzuteilen. Ist eine Ersetzung der evtl. vorhandenen medizinischen Abkürzungen erwünscht, wird auch diese in diesem Schritt vorgenommen.
Die Funktion wird per Kommandozeilenparameter „-prct“ bzw. „preprocess reference corpus text“ aufgerufen.

c) Reference Corpus Generation

Ziel dieser Funktion die Erzeugung des Referenzkorpus, welcher unter anderem die benötigten Häufigkeiten zur Differenzanalye beinhaltet.
Der Aufruf dieser Funktion erfolgt per Kommandozeilenparameter „- rcg“ bzw. „reference corpus generation“ aufgerufen. Nach Abschluss der Berechnung wird neben der dafür benötigten Zeit, die Sprache des Korpus, die Anzahl der gefundenen Wortformen sowie die verwendeten POS-Muster auf der Konsole ausgegeben.
Das Ergebnis dieser Funktion ist die Datei *referenceCorpus.dat* welche im Ordner „working“ gespeichert wird.

d) Analysis Corpus Generation

Ziel dieser Funktion ist es den Analysekorpus zu erzeugen, welcher unter anderem die benötigten Häufigkeiten zur Differenzanalye beinhaltet.
Die Funktion wird per Kommandozeilenparameter „- acg“ bzw. „analysis corpus generation“ aufgerufen. Auch hier wird nach Abschluss der Berechnung eine kurze Zusam-

menfassung des erzeugten Korpus auf der Konsole ausgegeben.
Das Ergebnis dieser Funktion ist die Datei *analysisCorpus.dat*, welche im Ordner „working“ gespeichert wird.

e) Terminology Extraction

Ziel dieser Funktion ist es, die Terminologie-Extraktion via Differenzanalyse und der nachfolgenden Analyse der gefundenen Fachterme auszuführen.
Die Funktion wird per Kommandozeilenparameter „- te <'minFreqText;minSig'>“ aufgerufen. Dabei steht “minFreqText“ für die Mindesthäufigkeit eines Terms im Analysekorpus und „minSig“ für die Mindestsignifikanz, die durch Berechnung Häufigkeitsquotienten erfolgt.
Nach Abschluss der Bearbeitung wird die dafür benötigte Zeit, die Anzahl der gefundenen Fachterme sowie die Anzahl an Knoten und Kanten, welche bei der Erweiterung des Graphen eingefügt werden, auf der Konsole ausgegeben.
Die Dateien „relevantTerms.csv“ und „droppedTerms.txt“ sind das Ergebnis dieser Funktion; beide befinden im Ordner „working“ des aktuellen Prorgammverzeichnisses.

f) Graph Extender

Diese Funktion dient dazu, die gefundenen Fachterme als Knoten in den gegebenen WikiGraphen einzupflegen und diese durch Kanten mit den bereits existierenden Knoten zu verbinden. Das Schreiben des erweiterten WikiGraphen auf die Festplatte. Diese Funktion wird per Kommandozeilenparameter „-ex“ beziehungsweise „-extend graph“ aufgerufen.
Als Ergebnis dieser Funktion wird ein erweiteter Graph im Ordner „result“ des aktuellen Programmverzeichnisses abgespeichert.
Teilschritt a) und b) müssen zuvor ausgeführt worden sein. Ist dies nicht der Fall, werden beide Teilschritte automatisch ausgeführt.

g) Hilfe

Bei dieser Funktion wird ein Hilfstext auf der Konsole mit den Möglichkeiten, welche das Tool zur Verfügung stellt, angezeigt. Die Auflistung beinhaltet eine Kurzbeschreibung der jeweiligen Funktionen, wobei der Aufruf des Hilfstextes per Kommandozeilenparameter „-h“ beziehungsweise „– help“ erfolgt.

Um das WikiGraph-TE-Tool flexibel zu gestalten, wird eine Property-Datei verwendet, mit deren Hilfe die Applikation konfiguriert werden kann. Eine Beschreibung dazu befindet sich im Anhang (vgl. Anhang A).

5 Ergebnisse

Dieses Kapitel umfasst die Ergebnisse, welche mit Hilfe des *WikiGraph-TE-Tool* produziert wurden.
Dabei wird zuerst eine Leistungsbetrachtung der einzelnen Teilschritte, die Ergebnisse der Terminologie-Extraktion sowie die Informationen zur Erweiterung des WikiGraphen dargestellt. Anschließend erfolgt die Evaluation der Ergebnisse und ein Leistungsvergleich zur ASV-Toolbox TE.
Zunächst werden jedoch allgemeine Informationen zur Ergebniserzeugung dargelegt.

Allgemeine Informationen

Die nachfolgenden Ergebnisse stützen sich auf folgende Gesundheitsgraphen

- Deutsch: healthgraph-german-jan-2012-SNAPSHOT-0.8.0
  Anzahl Konzept-Knoten: 101847
- Englisch: healthgraph-english-jan-2012-SNAPSHOT-0.8.0
  Anzahl Konzept-Knoten: 348541

Für die Erzeugung der Korpora wurden folgende POS-Muster verwendet:

- Deutsch: NN;ADJA NN;NN APPR NN;NN ART NN
- Englisch: NN;ADJA NN;NN NN;ADJA NN NN

Die Auswahl der eingesetzten POS-Muster basiert auf Grundlage der in Kapitel 2.3.5 vorgestellten Ansätze zur Terminologie-Extraktion.

5.1 Leistungsbetrachtung

Es folgt ein Überblick über die Performance des *WikiGraph-TE-Tools*.

Die Applikation wurde auf einer Virtual Machine mit Intel Xeon Dualcore[1] getaktet mit maximal 2,4 GHz und 16GB RAM ausgeführt, wobei zwei Rechenkerne zur Verfügung standen. Da es sich um eine virtuelle Infrastruktur im Produktivbetrieb des Studiengangs Medizinischer Informatik handelt, können die Zeiten zur Berechnung der einzelnen Schritte teilweise variieren.

[1] Intel(R) Xeon(R) CPU E5645 @ 2.40GHz

In Tabelle 5.1 sind die zur Berechnung der einzelnen Teilschritte benötigten Zeiten, sowie die daraus resultierende Gesamtzeit dargestellt.

	Deutsch		Englisch	
Abkürzungsersetzung	nein	ja	nein	ja
Referenz-Korpus-Erzeugung	211	215	84	88
Analysis-Korpus-Erzeugung	126	131	447	476
TE und Analyse	2	2	2	2
Erweiterung d. Graphen	1	1	1	1
Gesamtzeit	340	349	534	567

Tabelle 5.1: WikiGraph-TE-Tool: benötigte Zeiten der Teilschritte [Min]

Die Gesamtzeit varriert dabei zwischen ungefähr 6 Stunden für die deutsche und ca. 10 Stunden für die englische Sprache.
Bei der Verarbeitung der Referenzkorpus-Texte werden jeweils Textdateien, welche 10 Millionen Sätze enthalten, mit der Gesamtgröße von 1,4 Gigabyte, prozessiert. Die deutschen Rohtexte der Wikipedia haben eine Gesamtgröße von 624 Megabyte, die Rohtexte in der englischen Gesundheitsdomäne der Wikipedia besitzen eine Gesamtdateigröße von 3,3 Gigabyte. Die Satzanzahl der Analysekorpora wurden per SentenceDetector (vgl. Kapitel 2.3.2.1) ermittelt. Der deutsche Analysetext setzt sich aus 4.777.622 Sätzen zusammen, der englische Analysetext beinhaltet 22.654.189 Sätze. Die nachfolgende Tabelle veranschaulicht den Durchsatz während der Korpuserzeugung.

	Deutsch		Englisch	
Abkürzungsersetzung	nein	ja	nein	ja
Referenz-Korpus	786	755	1984	1894
Analysis-Korpus	632	608	845	793

Tabelle 5.2: WikiGraph-TE-Tool: Durchsatz während der Korpuserzeugungs [Sätze/Sec]

Der höchste Durchsatz mit 1984 Sätzen pro Sekunde wurde bei der Erzeugung des englischen Referenzkorpus ohne Abkürzungsersetzung erreicht. Im Deutschen betrug der höchste Durchsatz 789 Sätze pro Sekunde. Dieser Wert wurde ebenfalls bei der Erzeugung des Referenzkorpus ohne Abkürzungserzeugung aufgestellt. Bei der Erstellung des Analysekorpus schwankt der Durchsatz für die deutsche Sprache zwischen 608 Sätzen pro Sekunde mit Abkürzungsersetzung und 632 Sätzen pro Minute ohne Abkürzungsersetzung; in der englischen Sprache zwischen 793 Sätzen pro Minute und 845 Sätzen pro Minute.

5.2 Ergebnisse des WikiGraph-TE-Tools

Bei der Terminologie-Extraktion und Erweiterung des Gesundheitsgraphen für die deutsche und englische Sprache wurden die in Tabelle 5.3 dargestellen Ergebnisse geliefert. Die Ergebnisse wurden mit der Mindestfrequenz im Analysekorpus von 2 und dem HQ von 500 als Schwellwerte erzeugt. Es sind die Anzahl der gefundenen Fachterme, die Anzahl der bereits im Gesundheitsgraphen vorhandenen Knoten, die Anzahl der duch die Erweiterung hinzugefügten Knoten, sowie die Anzahl der neuen Kanten aufgelistet.

	Deutsch		Englisch	
Abkürzungsersetzung	nein	ja	nein	ja
gef. Fachterme	747	1027	289	460
existierende Knoten	314	372	113	156
neue Knoten	433	655	176	304
neue Kanten	291.619	369.264	285.935	519.438

Tabelle 5.3: WikiGraph-TE-Tool: Ergebnisse (minFreq = 2 ; minHQ = 500)

Es wurden durch die Terminologie-Extraktion 747 potentielle Fachterme für die deutsche Sprache (ohne Abkürzungsersetzung) gefunden. Davon waren bereits 314 Fachterme als Knoten im Gesundheitsgraphen vorhanden. Bei der Erweiterung des Gesundheitsgraphen werden 433 neue Knoten und 291.619 neue Kanten eingefügt. Ohne Abkürzungsersetzung wurden für die englische Sprache 289 Fachterme gefunden. Davon waren bereits 113 im englischen Gesundheitsgraphen vorhanden, weshalb eine Erweiterung des Graphen um 176 Knoten und 285.935 Kanten stattfinden konnte.
Durch Abkürzungsersetzung ist die Anzahl der extrahierten Terme in beiden Sprachen angestiegen. Der Zuwachs beträgt für die deutsche Sprache 37%. Im Englischen erhöhte sich die Anzahl der Terme von 289 auf 460, welches einer Zunahme von 59,2% entspricht.

Es folgt ein Ausschnitt der durch die Terminologie-Extraktion gefundenen Fachterme der deutschen Sprache ohne Abkürzungsersetzung (absteigend sortiert nach HQ).

Rang	Fachterm	HQ	HKL
1	Einzelnachweise	37939.85	6
2	Musculus	18201.75	8
3	Summenformel	17633.35	8
4	GefStKz	16644.76	8
5	PubChem	15247.22	9
...			
100	Citratzyklus	1360.02	11
101	Strukturhinweis	1354.39	12
...			
500	Mediastinum	609.66	13
501	Infektiöse	607.78	13
...			
747	Vitalkapazität	500.86	13

Tabelle 5.4: Ausschnitt: Deutsche Fachterme (minFreq = 2 ; minHQ = 500)

Die nachfolgende Tabelle zeigt einen Ausschnitt für die englische Sprache ohne Abkürzungsersetzung (absteigend sortiert nach HQ):

Rang	Fachterm	HQ	HKL
1	bioavailability	6163.28	9
2	molecular_weight	4440.42	9
3	tradename	4286.21	8
4	routes_of_administration	4199.33	9
5	ester	3072.39	10
...			
100	lymphocyte	784.32	12
101	conjunctiva	784.32	12
...			
200	urethral	602.19	12
201	valproate	599.75	12
...			
289	cyncobolamin	500.31	12

Tabelle 5.5: Ausschnitt: Englische Fachterme (minFreq = 2 ; minHQ = 500)

Nachdem die durch Differenzanalyse gefundenen Fachterme extrahiert wurden, erfolgt eine Analyse der Fachterme. Dabei wird überprüft, ob ein Fachterm bereits als Konzept im Gesundheitsgraphen vorhanden ist. Die nachfolgende Tabelle zeigt einen Ausschnitt aus der *droppedTerms.txt*, welche diese aussortierten Terme enthält.

deutsch	englisch
PubChem	routes_of_administration
Gefahrensymbole	lindane
Therapeutisches Verfahren	elimination_half-life
Ligamentum	reductase
Freiname	hypersexuality
Ganglion	axon
Pathogenese	epithelium
Glykolyse	keratosis
Resorption	cytosol
Prokaryoten	chloramphenicol
Atmungskette	meprobamate

Tabelle 5.6: Auschnitt aus der engl. und dt. droppedTerms.txt

Fachterme, wie z.B. „Ganglion“ oder „Atmungskette“, werden also nicht als neue Knoten zu der Erweiterung des Gesundheitsgraphen herangezogen.

Befindet sich ein gefundener Fachterm nicht im existierenden Gesundheitsgraph, werden diese inklusive der Artikelnamen, in denen sie aufgefunden wurden, in die Datei *relevantTerms.csv* geschrieben. Die nachfolgende Abbildung zeigt einen Ausschnitt der für den englischen Gesundheitsgraphen erzeugten Datei.

gefundener Fachterm	Artikel, in denen FT vorkommt			
bioavailability	Landruma	Cefuroxime_axetil	Zinc L-aspartate	...
ester	C11_H17_O2_N2_S_Na	Sport_supplements	Lophobios	...
hydrolysis	Aquarex_methyl	Diethyleneglycol	Adenylate_charg	...
reticulum	Insulin_zinc_susp_prompt_bee	Thesaurocytes	Neurone	...
ligamentum	Double aortic arch	Aortic_Dissection	Intrauterine	...
isomer	Acetylmethadol	Gamma_benzene_hex	4,5-methylenedioxy-3-methylamphetamine	
cofactor	ATC_code_A01AD01	Levorenin	Adrenamine	....
glycine	Serenamin	Insulin_zinc_susp_pro	Trilon_BS	...
moiety	Cefuroxime_axetil	Polyubiquitination	Colitis,_ulcerativ	...
anion	Trilon_BS	Neurone	PFOA	...
heme	List of MeSH codes (D04)	Erythroid	B6_Vitamin	...
protein_bound	Sostril	Urbol	C26H28Cl2N4O4	...
disulfide	Insulin_zinc_susp_prompt_bee	Redox-sensitive_Gree	Octyldodecanol	...
action potential	Calcium_ion	Neurone	Xylocaine	...

Abbildung 5.1: Ausschnitt der englischen relevantTerms.csv (in Excel-Dartellung)

Beispielsweise ist der englische medizinische Fachterm „anion“ unter anderem in den Artikeln „Trilon_BS“, „Neurone“ und „PFOA“ aufzufinden. Der Fachterm „action potential“ ist ebenfalls im Artikel „Neurone“ sowie in den Artikeln „Calcium_ion“ und „Xylocaine“ vorhanden.

5.3 Evaluation

Um ein System zu evaluieren, werden im Information Retrieval die wichtigen Maßzahlen *Precision* und *Recall* angewandt.
Folgende Größen dienen der Erläuterung dieser Maßzahlen:

- A = Anzahl richtig erkannter Fachterme
- B = Anzahl falsch erkannter Fachterme
- C = Anzahl nicht gefundener Fachterme

Die Wahrscheinlichkeit mit der es sich bei einem durch das WikiGraph-TE-Tool gefundenen Fachterm auch wirklich um einen Fachterm handelt, wird *Precision* genannt. Zur Bestimmung dieses Maßes wird die Anzahl der gefundenen Fachterme (richtig oder falsch erkannt) sowie die Anzahl der nur richtig erkannten Fachterme benötigt. Die *Precision* kann durch nachfolgende Formel berechnet werden [18]:

$$Precision = \left(\frac{A}{A+B}\right) \tag{5.1}$$

Als *Recall* würde die Wahrscheinlichkeit deklariert werden, dass ein Fachterm tatsächlich als Fachterm erkannt wird. Diese berechnet sich durch [18]:

$$Recall = \left(\frac{A}{A+C}\right) \tag{5.2}$$

Da es jedoch bei der großen Anzahl an Fachtermen in den Rohtexten der Wikipedia nicht möglich ist, jeden Fachterm manuell zu identifizieren, kann in diesem Fall nur das Maß *Precision* zur Evaluation eingesetzt werden.

Die nachfolgenden Werte zur *Precision* der Terminologie-Extraktion wurden händisch verifiziert, da keine vollständige Liste an medizinischen Fachtermen vorhanden ist, welche zur Überprüfung der Korrektheit der Fachterme herangezogen werden kann. Eine vollständige Auflistung der englischen Terme ohne Abkürzungsersetzung befindet sich in Anhang B. Die Liste der deutschen Terme ohne Akürzungsersetzung ist in Anhang C aufgelistet.

	Deutsch		Englisch	
Abkürzungsersetzung	nein	ja	nein	ja
Anzahl gefundener FT	744	1027	289	460
korrekt erkannte FT	711	925	282	441
falsch erkannte FT	33	102	7	19
Precision	0,96	0.90	0,98	0,96

Tabelle 5.7: WikiGraph-TE-Tool: Precision (minFreq:2 ; minHQ 500)

Da der Umfang der durch den Schwellwert erhaltenen Fachterme im Englischen nicht repräsentativ ist, wurden die Auswertungen auf eine, noch manuell durchführbare Analyse, von tausend Fachtermen erweitert:

	Deutsch		Englisch	
Abkürzungsersetzung	nein	ja	nein	ja
korrekt erkannte FT	956	906	973	964
falsch erkannte FT	44	94	27	36
Precision	0,96	0,91	0,97	0,96

Tabelle 5.8: WikiGraph-TE-Tool: Precision der ersten 1000 Terme

Die Precision ist für die deutschen Fachterme ohne Abkürzungsersetzung während der Korporaerzeugung konstant geblieben. Mit Abkürzungsersetzung hat sich die Precision um 1% gesteigert (da 27 Terme weniger der Analyse unterzogen wurden).
Für die Fachterme der englischen Sprache hat sich die Precision auf 97% verringert. Mit Abkürzungsersetzung ergab sich ein Wert von 96%.

5.4 Vergleich zur ASV-Toolbox

Wie bereits in Kapitel 1.3 angesprochen, wurde in einer vorangestellten Studienarbeit [15] die Fremdbibliothek „ASV-Toolbox-TE“ [29] zur Terminologie-Extraktion eingesetzt. Es folgt ein Vergleich der Leistung der im Rahmen dieser Arbeit entwickelten Komponente zur Terminologie-Extraktion und der Implementierung der ASV-Toolbox.

Die Gegenüberstellung wurde auf einem Intel(R) Core 2 Duo(R) getaktet mit 2.0 GHz und 4GB RAM durchgeführt. Dabei wurden die Gesamtzeit des Terminoligie-Extraktion

und des nachfolgenden Schritts der Analyse der gefundenen Terme gemessen. Die Zeitmessung wurde, sobald die Wikipedia-Artikel auf der Festplatte vorlagen, gestartet.

In Tabelle 5.9 ist die bei der Messung verarbeitete Artikelanzahl sowie die gemessenen Zeit dargestellt.

Anzahl Artikel	ASV-Toolbox TE	WikiGraph-TE-tool
1	0,5	1,3
10	1,5	1,3
100	3,6	1,5
1.000	23	4
10.000	x	24

Tabelle 5.9: Vergleich ASV-Toolbox TE vs. WikiGraph-TE-Tool [min]

Ab einer Anzahl von über 100 verarbeiteten Artikeln wird der Unterschied der Bearbeitungsgeschwindigkeit zwischen der ASV-Toolbox und des WikiGraph-TE-Tools deutlich. Während die Komponente der ASV-Toolbox für die Prozessierung von 1.000 Artikeln 23 Minuten benötigt, wurden die identischen Artikel vom WikiGraph-TE-Tool in 4 Minuten verbeitet; dies entspricht einer Zeitersparnis von 83,6%. Für die Verarbeitung von 10.000 Artikeln benötigt das WikiGraph-TE-Tool 24 Minuten. Die Messung der ASV-Toolbox TE wurde nach 3 Stunden abgebrochen. Die nachfolgende Abbildung 5.2 veranschaulicht die durchgeführte Testreihe graphisch.

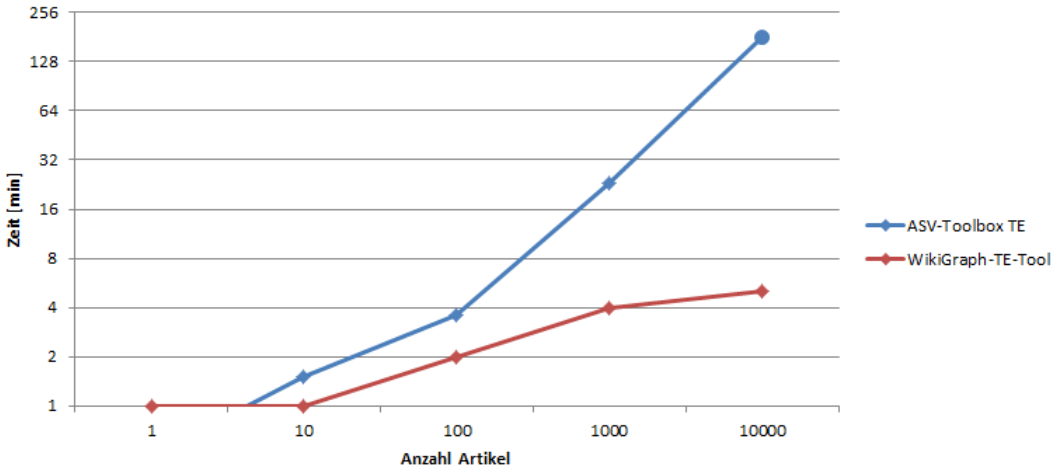


Abbildung 5.2: Leistungsvergleich: ASV-Toolbox TE vs. WikiGraph-TE-Tool

6 Diskussion und Ausblick

In diesem Abschnitt werden die in Kapitel 5 aufgezeigten Ergebnisse kritisch betrachtet. Danach folgt ein Ausblick über die weitere Verwendung des WikiGraph-TE-Tools.

6.1 Diskussion der Ergebnisse

Die nachfolgende Aufzählung soll einen Überblick über die im Rahmen der Arbeit umgesetzen Anforderungen und erreichten Ziele geben:

- Erzeugung eines geeigneten Referenz- und Analysekorpus
- Eigenimplementierung einer Differenzanalyse zur Extraktion von Fachtermini
- Realisierung einer Softwarelösung zur Erweiterung des bestehenden Gesundheitsgraphen

Ziel dieser Arbeit war es, eine Softwarelösung zu entwickeln, welche einen bestehenden Gesundheitsgraphen um bisher nicht in den regulären Strukturen abgebildeten „Assoziationen“ erweitert. Dieses Ziel wurde erreicht. Ausgehend von der Differenzanalyse zur Extraktion von Fachtermini können bestehende Gesundheitsgraphen um die gefundenen Fachterme und deren Verlinkung in bereits extistierende Konzepte erweitert werden.

Die Erzeugung der für die Differenzanalyse grundlegenden Referenz- und Analysekorpora ist performant (vgl. Tabelle 5.2). Die anschließende Terminologie-Extraktion via Differenzanalyse und die danach angereihte Analyse der gefundenen Fachterme wird ebenfalls in sehr kurzer Laufzeit ausgeführt (vgl. Tabelle 5.1). Im Vergleich zur exisstierenden Softwarelösung „ASV-Toolbox TE“ wurde ein klarer Leistungsvorsprung nachgewiesen (vgl. Kapitel 5.4).

Korpora, welche nicht durch Abkürzungsersetzung von medizinischen Fachteremen verändert wurden, weisen die Ergebnismengen der Differenzanalyse in dem analysierten Ausschnitt eine sehr hohe Präzision auf. Die Präzision beträgt 96% in der deutschen Sprache und 97% in der englischen Sprache (vgl. Tabelle 5.8). Dieses Ergebnis wurde trotz technischer Begriffe, welche in den Rohtexten der Wikipedia und der Ergebnismenge der Terminologie-Extraktion vorhanden sind und sich negativ auf die Präzision auswirken, erziehlt.Trotz Vorfilterung der Rohtexte sind technische Begriffe wie zum Beispiel „style=background-color“ oder „url=http“ weiterhin enthalten.
Durch Ersetzung der in den Korpustexten enthaltenen medizinischen Abkürzungen durch

ihre Vollschreibweise hat sich die Precision auf 90% für die deutsche Sprache verschlechtert. Das Ersetzen der medizinischen Fachterme wirkte sich bei der englischen Sprache weniger stark auf die Precision aus. Sie nahm um 1% ab. Es ist nicht ausgeschlossen, dass derartige Ersetzungen das Ergebnis in seiner Präzision verzerren können. Dies müsste daher getrennt bezüglich der Effektgröße untersucht werden. Die Ersetzung ist stark kontextabhängig. Die Abkürzung „IR“ des medizinischen Fachterms „Innenrotation„ wird in anderen Kontexten mit den Termen „Information Retrieval“„Indian Railways“, „Infantrieregiment“, „Infrarot“, ïnterne Revision“, „Interregio“, „Investor Relations“ oder „Iran“ abgekürzt.

Die Analyse der gefundenen Fachterme, welche zur Erweiterung des Gesundheitsgraphen dient, liefert interessante Ergebnisse. Ein Beispiel hierzu findet sich im englischen Gesundheitsgraphen. Es wurden die Fachtermini „anion“ und „action potential“ extrahiert. Beide medizinischen Fachterme treten im bereits vorhandenen Konzept-Knoten „Neurone“ auf. Somit wurde eine bisher unstrukturierte Assoziation gefunden, welche Fachkundigen klar ist, aber noch nicht in dem vorhandenen Graphen abgebildet war.

6.2 Ausblick

Das WikiGraph-TE-Tool könnte in Zukunft auf anderen Domänen der Wikipedia, wie zum Beispiel der Informatik oder der Wirtschaft, angewandt werden. Durch den vorhandenen Referenzkorpus in Kombination eines Analysekorpus, welcher durch Rohtexte der Wikipedia dargestellt wird, könnten mit Hilfe der Differenzanalyse Fachterme jedes Fachgebiets extrahiert werden.
Die Einbindung weiterer Sprachen wäre ebenfalls denkbar, sofern ein geeigneter Referenzkorpus und eine geeignetes Tool oder Verfahren zur Grundformreduktion vorhanden ist.
Da der Umfang der Wikipedia immer weiter zunimmt, wäre es sinnvoll, das Einlesen der Rohtexte so umzuschreiben, dass die aufgesplitteten Rohtext-Dateien erst eingelesen werden, wenn sie benötigt und nicht, wie in der aktuellen Implementierung, auf einmal in den Speicher geladen werden.
Bei der Weiterverwendung des WikiGraph-TE-Tools ist zu überlegen, ob der Lemmatizer für die deutsche Sprache um weitere Wortformen erweitert oder durch ein anderes Verfahren zur lexikalischen Grundformreduktion ausgetauscht wird. Es können aktuell nicht alle gefundenen Wortformen auf die Grundform reduziert werden. Dies hat zur Folge, dass die Worthäufigkeiten der Grundform nicht richtig gezählt werden und sich somit negativ auf das Ergebnis der Differenzanalyse auswirken
Durch die genutzte Modularisierung (vgl. Kapitel 3.2) kann das Tool einfach in andere Applikationen integriert werden, wie z.B. das in TULUM angestrebte Empfehlungssystem.

Literaturverzeichnis

[1] Apache Foundation. Apache Maven. [Online], [Stand: 20.01.2012]. URL `http://maven.apache.org/`. [Last checked: 05.02.2012].

[2] Apache Foundation. Apache OpenNLP. [Online], [Stand: 28.11.2011]. URL `http://incubator.apache.org/opennlp/`. [Last checked: 03.01.2012].

[3] Apache Foundation. Apache OpenNLP Developer Documentation. [Online], [Stand: 28.11.2011]. URL `http://incubator.apache.org/opennlp/documentation/1.5.2-incubating/manual/opennlp.html`. [Last checked: 14.01.2012].

[4] Apache Foundation. Models for OpenNLP. [Online], [Stand: 04.05.2011]. URL `http://opennlp.sourceforge.net/models-1.5/`. [Last checked: 13.01.2012].

[5] Antti Arppe. Term Extraction from Unrestricted Text. [Online], [Stand: 23.11.2005]. URL `http://www2.lingsoft.fi/doc/nptool/term-extraction.html`. [Last checked: 17.01.2012].

[6] Norbert Bensch and Michael Stetter. Pluralbildung. [Online], [Stand: 31.01.2012]. URL `http://www.mein-deutschbuch.de/lernen.php?menu_id=53#pluralformen`. [Last Checked: 05.02.2012].

[7] Joshua Bloch. *Effective Java (2nd Edition): A Programming Language Guide*. Addison-Wesley Longman, Amsterdamm, 2 edition, 2008. ISBN 978-0321356680.

[8] Abraham Bookstein and Don R. Swanson. Probabilistic models for automatic indexing. *Journal of the American Society for Information Science*, 25(5):312ff, Sep/Oct 1974.

[9] Didier Bourigault. Surface grammatical analysis for the extraction of terminological noun phrases, August 1992. 977-981 pp.

[10] Kai-Uwe Carstensen, Christian Ebert, and Cornelia Ebert. *Computerlinguistik und Sprachtechnologie: Eine Einführung*, volume 3. Auflage. Spektrum Akademischer Verlag Heidelberg, 2010. URL `http://books.google.de/books?id=b9AvBCULhkYC&printsec=frontcover&hl=de&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false`.

[11] Universität München Centrum für Informations-und Sprachverarbeitung. Was ist das CIS? [Online], [Stand: 23.11.2009]. URL `www.cis.uni-muenchen.de/download/flyer/cis_flyer.pdf`. [Last checked: 17.01.2012].

[12] Beatrice Daille, Eric Gaussier, and Jean-Marc Langé. Towards atomatic extraction of monolingual and bilangual terminology, 1994. 512-521 pp.

[13] Fred Damerau. Generating and evaluating domain-oriented multi word terms from texts. *Information Processing and Management*, 29(4):443–447, 1993.

[14] H.-R. Fulck. Fachsprachen.de. [Online], [Stand: Mai 2002]. URL `http://www.fachsprachen.de/`. [Last checked: 20.01.2012].

[15] Heiko Gramlich. WikiGraph-TE-Tool - Eine Anwendung zur Terminologie-Extraktion aus Wikipedia-Artikeln zur Erweiterung von Gesundheitsgraphen. Master's thesis, Hochschule Heilbronn, 2011.

[16] Ulrich Heid. A linguistic bootstrapping approach to the extraction of term candidates from german text. *Terminology*, 5(2):161–181, 1998.

[17] Gerhard Heyer, Uwe Quasthoff, and Thomas Wittig. *Text Mining: Wissensrohstoff Text - Konzepte, Algorithmen, Ergebnisse*. W3L GmBH, 2006. ISBN 978-3937137308.

[18] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Print, 1999. ISBN 0-262-13360-l.

[19] Praharshana Perera and René Witte. A Self-Learning Context-Aware Lemmatizer for German. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, pages 636–643. Association for Computational Linguistics, Vancouver, British Columbia, Canada, October 6–8 2005. URL `http://www.aclweb.org/anthology/H/H05/H05-1080`.

[20] Martin Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, Juli 1980.

[21] G. Salton, A. Wong, and C. S. Yang. A vector soace model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[22] K. Schmalenbach. Terminologie - Grundbegriffe. [Online], [Last checked: 07.12.2001]. URL `http://www.doku.net/artikel/terminolo1.htm`. [Stand: 21.12.2012].

[23] Klaus-Dirk Schmitz and Sonia Kakonen. Fachsprache vs. Gemeinsprache. [Online], [Stand: 21.11.2011]. URL `http://www.termportal.de/elearning/03_fachsprache.html`. [Last checked: 21.01.2012].

[24] TechTerms.com. Definition Repository. [Online], [Stand: 18.08.2011]. URL `http://www.techterms.com/definition/repository`. [Last checked: 13.01.2012].

[25] A. J. Thomson and A. V. Martinet. *A Practical English Grammar*. Oxford University Press, 4. edition, 1986. ISBN 0194313425.

[26] Xia Tian and Wang Tong. An improvement to tf: Term distribution based term weight algorithm, 2010. ISBN 978-0-7695-4011-5. 252-255 pp.

[27] Benjamin Trinczek. WikiGraph Analyse, Extraktion und Abbildung von medizinischen Strukturmerkmalen der Wikipedia durch Realisierung eines flexiblen Graphen- Frameworks. Master's thesis, Hochschule Heilbronn, 2010.

[28] Universität des Saarlandes. Negra corpus. [Online], [Stand: 11.05.2006]. URL `http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/negra-corpus.html`. [Last checked: 16.01.2012].

[29] Universität Leipzig. ASV Toolbox. [Online], [Stand: 25.01.2011]. URL `http://wortschatz.uni-leipzig.de/~cbiemann/software/toolbox/index.htm`. [Last checked: 01.01.2012].

[30] Universität Leipzig. Downloadseite des Referenzkorpus. [Online], [Stand: 12.08.2011]. URL `http://corpora.informatik.uni-leipzig.de/download.html`. [Last checked: 12.08.2011].

[31] Universität Tübingen. Informationsblatt: Stuttgart-Tübingen tagset (stts). [Online], [Stand: 09.12.2007]. URL `http://www.sfb441.uni-tuebingen.de/a5/codii/info-stts-de.xhtml`. [Last checked: 01.01.2012].

[32] J. Vollmert. *Grundkurs Sprachwissenschaft*. Fink (UTB), 5. edition, 1985. ISBN 3-8252-1879-1.

[33] Martin Wiesner and Daniel Pfeifer. TULUM: Text-based unique linguistic utilities in medicine. [Online], [Stand: 07.01.2012]. URL `http://www.hs-heilbronn.de/574903/tulum`. [Last checked: 07.01.2012].

[34] Wikikipedia. Wikipedia. [Online], [Stand: 17.12.2011]. URL `http://de.wikipedia.org/wiki/Wikipedia`. [Last checked: 18.12.2011].

[35] Wikimedia Foundation. List of wikipedias. [Online], [Stand: 15.12.2011]. URL `http://meta.wikimedia.org/wiki/List_of_Wikipedias`. [Stand: 17.01.2012].

[36] Wikimedia Foundation. Wikipedia. [Online], [Stand: 18.12.2011]. URL `http://www.wikipedia.org/`. [Stand: 18.12.2012].

[37] Wikimedia Foundation. Wikipedia - Wikitext. [Online], [Stand: 12.11.2011]. URL `http://de.wikipedia.org/wiki/Wikitext`. [Last checked: 18.12.2011].

[38] Wikipedia. Häufigkeitsklasse. [Online], [Stand: 03.10.2011]. URL `http://de.wikipedia.org/wiki/H%C3%A4ufigkeitsklasse`. [Last checked: 03.01.2012].

[39] Wikipedia. Information retrieval. [Online], [Stand: 11.01.2012]. URL `http://de.wikipedia.org/wiki/Information_Retrieval`. [Last chekced: 16.01.2012].

[40] H.F. Witschel. Text, Wöter, Morpheme - möglichkeiten einer automatischen Terminologie-Extraktion. In *Proc. of GLDV-Tagung 2005*, pages 659–672, 2005.

[41] René Witte. The durm german lemmatizer. [Online], Februar 2006. URL `http://www.semanticsoftware.info/system/files/DurmLemmatizer-1.0.tgz`. [Last checked: 22.12.2011].

[42] René Witte and Jutta Mülle. Text mining: Text-mining: Wissensgewinnung aus natürlichsprachigen Dokumenten. Technical report, Universität Karlsruhe (IPD), 2005. URL `digbib.ubka.uni-karlsruhe.de/volltexte/documents/3230`.

A Parametrisierung des WikiGraph-TE-Tools

Inhalt der Konfigurationsdatei *wikigraph-te-tool-config.properties*:

```
!General settings
!Locale (de/en)
locale = de
!------
!Corpus settings
!--------------------
!DE: deu_news_2011_10M-sentences.txt
!EN: eng_news_2010_10M-sentences.txt
cg.referenceCorpusTextFilename = deu_news_2011_10M-sentences.txt
cg.analysisCorpusTextFilenameSuffix = _wikipediaRawText.txt
!-----
!corpus generation
!DE: die
!EN: the
cg.mostFrequentWord = die
!-----
cg.useStemming = true
!----------
cg.useAbrevReplacement = true
cg.prefixAbbreviationsReplaced = abbreviationsReplaced_
cg.prefixCleanedText = cleaned_
cg.fileSplittingSizeInMB = 100
!@see corpora-api#WordFormPatterns for all possible patterns
!DE: NN;ADJA NN;NN APPR NN;NN ART NN
!EN: NN;ADJA NN;NN NN;ADJA NN NN
cg.usedWordFormPatterns = NN;ADJA NN;NN APPR NN;NN ART NN
cg.referenceCorpusTextFileIsCompressed = false
cg.analysisCorpusTextFileIsCompressed = false
cg.referenceCorpusFileNameSuffix = _referenceCorpus.dat
cg.analysisCorpusFileNameSuffix = _analysisCorpus.dat
!--------------------
!GraphExtender settings
!DE: Gesundheit
!EN: Medicine
ex.extensionFileName = relevantTerms.csv
ex.wikiGraphDomainName = Gesundheit
!--------------------
```

Erläuterung der einzelnen Parameter:

locale	Sprache, welche verwendet werden soll
cgg.referenceCorpusTextFilename	Dateiname des Referenzkorpustext
cg.analysisCorpusTextFilenameSuffix	Suffix des Analysekorpustext Dateinamen
cg.mostFrequentWord	das häufigste Wort der Sprache
cg.useStemming	Grundformreduktion aktiv oder nicht
cg.useAbrevReplacement	Abkürzungsersetzung aktiv oder nicht
cg.prefixAbbreviationsReplaced	Prefix, welches bei aktiver Abkürzungsersetzung der vorverarbeiteten Referenzkorpustext-Datei angefügt wird
cg.prefixCleanedText	Prefix, welches nach der Vorverarbeitung des Referenzkorpustextes an die Textdatei angefügt wird
cg.fileSplittingSizeInMB	gewünschte Dateigröße der Korpustextdateien
cg.usedWordFormPatterns	verwendete POS-Muster
cg.referenceCorpusTextFileIsCompressed	Referenzkorpustext ist komprimiert oder nicht
cg.analysisCorpusTextFileIsCompressed	Analysekorpustext ist komprimiert oder nicht
cg.referenceCorpusFileNameSuffix	Suffix der Referenzkorpusdatei
cg.analysisCorpusFileNameSuffix	Suffix der Analysekorpusdatei
ex.extensionFileName	Dateiname der zur Erweiterung des Graphen erstellten Datei
ex.wikiGraphDomainName	Domänenname des Graphen

B Ergebnis der TE für die englische Sprache

In nachfolgender Tabelle ist das Ergebnis der Terminologie-Extraktion für die englische Sprache ohne Abkürzungsersetzung abgebildet (minFreq: 2, minHQ: 500). Falsch erkannte Terme wurden *kursiv* markiert.

bioavailability	molecular_weight	tradename
routes_of_administration	ester	hydrolysis
classis	binomial_authority	lindane
elimination_half-life	reductase	hypersexuality
axon	plant pathogen	zh-yue
epithelium	keratosis	cerebellar
reticulum	subclassis	cytosol
ligamentum	*image_caption*	berk
zh-min-nan	nevu	isomer
hydroxyl	cofactor	chloramphenicol
meprobamate	carbamazepine	prodrug
disease virus/paramyxovirus	virus/paramyxovirus	palmoplantar
glycine	hypertrichosis	glycolysis
moiety	chuan	*bat-smg*
celecoxib	anion	maxillary
ligament	heme	protein_bound
disulfide	action potential	porphyria
nystagmus	pneumothorax	lauryl sulfate
thalamus	phenytoin	pregnancy_category
ductus	palmar	depolarization
myocardium	vasoconstriction	reabsorption
acetyl	ramus	noradrenaline
synthase	acid cycle	metabolite
regnum	fluoroquinolone	polymerization
catechin	phagocytosis	ultrasonography
superfamily	substantia	carbonyl
distemper virus	medullary	guaifenesin
acetylation	superoxide	phenylbutazone
nerve n	blanket sleeper	syn
mesoderm	bronchitis virus	ethylene oxide
dermatosis	aldehyde	decarboxylase
purine	gluconeogenesis	metronidazole
levodopa	bilayer	methylene
lymphocyte	conjunctiva	tunica
aldosterone	cox-2	xylem
amoxicillin	diazepam	acid image
capitis	respiratory depression	resorption
midbrain	orthography	omeprazole

pruritus
receptor antagonist
transferase
zona
parainfluenza
cyanosis
glutamine
image_width
glycoprotein
anticoagulation
phenacetin
bursal
disulfiram
hemolysis
gumboro disease
scopolamine
hypertrophy
kcl
cyclooxygenase
histidine
hyperkalemia
subunit
anhydrase
excretion
caudal
cell adhesion
pleura
stroma
language=german
diethyl
eggman
ch'uan
ventral
catabolism
digraph
tuberosity
temazepam
chorea
acetylcholine receptor
phenethylamine
ectoderm
pallidum
parkinsonism
parenchyma
zopiclone
unranked_classis
lorazepam
oxazepam
prednisolone
birth_place
carcinogenicity
redox
elimination half-life
trigraph
status epilepticus
hypotonia
biosynthesis
bursal gumboro
alkyl
minora
parainfluenza virus
phenylephrine
dapsone
overdosage
arteriosus
lamina
subclavian
pathophysiology
anastomosis
diuresis
reference range
ataxia
urethral
conjugation
tracheal intubation
mycoplasma toxoid
tubercle
protein
side chain
arteritis
uveitis
endocytosis
spironolactone
heart block
cation
methotrexate
sagittal
methylphenidate
ileum
membrane potential
panniculitis
hepatotoxicity
oxidation state
peritoneum
palatine
secretory
epilepticus
dehydrogenase
polypeptide
gingival
luteum
gumboro
synthetase
toxoid
coronavirus
polyneuropathy
accessdate
rhinotracheitis
pharyngitis
leptospira
death_place
parietal
meatus
dysmenorrhea
acceptor
plasma membrane
valproate
erythema
glycosylation
transaminase
levofloxacin
hydroxyl group
mescaline
albinism
intrauterine
rhinotracheitis virus
cimetidine
wolfberry
efferent
myoclonus
emedicine
sah

pudendal
insufflation
storage disease
contrecoup
ductu arteriosu
pyrophosphate
romanization
histopathology
vitalism
prostaglandin
estradiol
labia minora
melting_point
mononitrate
cyanocobalamin
birth_date
syndrome type
gene transcription
sodium salt
diacetylmorphine
branch ramus
integrin
pathogenicity
carboxyl
cytochrome
carnitine
ichthyosis
legal_status
pupillary
pilocarpine
digitorum
symphysis
bibcode
substantia nigra
meristem
estrone
paracetamol
peptidoglycan
amine
atresia
sodium channel
human experimentation
hydroxyzine

C Ergebnis der TE für die deutsche Sprache

In nachfolgender Tabelle ist das Ergebnis der Terminologie-Extraktion für die deutsche Sprache ohne Abkürzungsersetzung abgebildet (minFreq: 2, minHQ: 500). Falsch erkannte Terme wurden *kursiv* markiert.

Einzelnachweise	Musculus	Summenformel
GefStKz	PubChem	Gefahrensymbole
Deutsche Geboren	GHS-Piktogramme	Arteria
ATC-Code	Dampfdruck	*Abgerufen*
Therapeutisches Verfahren	Akademischer Verlag	Eukaryoten
Ligamentum	GHS-Kz	Freiname
Ganglion	Pathogenese	Glykolyse
Löslichkeit	Flammpunkt	Molare
Monopräparate	Resorption	Prokaryoten
Andere Namen	Foramen	Strukturformel
ISBN	Atmungskette	Plexus
Chemische Eigenschaft	Suppl	Ipomoea
P-Sätze	Kristallsystem	CAS-Nummer
Anorganischen	R-Sätze	deutscher Medizin
Arzneistoff	Personenlexikon	Stratum
Cytoplasma	Gestorben	Strabismus
Granulozyten	Substituenten	Physikalische Eigenschaft
Innervation	Hyperthyreose	Gefahrstoff-kennzeichnung
Verschreibungspflicht	*Diagnostisches*	Organellen
Phosphorylierung	Kardiomyopathie	Benzolring
Meiose	Konstitutionsisomere	Biol
GHS-Einstufung	PMID	Allel
Biosynthese	Mitose	Raumgruppe
Enke-Verlag	unterschiedliche Anordnung	Explosionsgrenze
Volltextzugriff	Liganden	Ethen
Mykobakterien	Racemat	Letalität
Hydroxygruppe	*Franz-Viktor*	Sekretion
Archaeen	μm	Signaltransduktion
"Am J	*freier textzugriff*	Kapsid
Hyperparathyreoidismus	Proteinurie	Plantarum
Kohlhammer	Ramus	Alkoholen
Ordnungszahl	Mikrotubuli	Berlin Blau
Weiterführende Literatur	Handwörterbuch	Citratzyklus
Strukturhinweis	Endothelzellen	Gruyter
Basalmembran	Allele	Bauchwand
Staubblätter	Nävi	H_2O
Kniegelenkes	*"The*	Pfortader
Genlocus	DrugBank	Ruptur

Epithelzellen
Plasmahalbwertszeit
Verbandkasten
Musculi
Psychiatry
Nervus
Stoffgruppe
RNA-Polymerase
Kronblätter
Tachykardie
Feststoff
.Vgl
Gameten
Vorhersagewert
Fernmetastasen
Schweizer Geboren
Pharmakotherapie
Juniperus
Hypothyreose
Hämostase
Veresterung
Deletion
Virushülle
Adduktion
Iod
Enantiomere
.a.
Hypoxie
Lysosomen
Translokation
Interleukin-1
Menisken
Arztebl
Hodens
Fluorwasserstoff
Symphytum
klinischen Bild
Aminogruppe
Differentialdiagnose
höchsten Schmelzpunkt
Inhibitoren
Standardbildungsenthalpie
Sicherheitsdatenblatt
Schwefeltrioxid
Vesikel
Schmelzpunkt
KHK
Digitalisierte Ausgabe
Weblinks
Autoantikörper
Reposition
Erregungsleitung
Isoformen
Pharmazeutischer
Immunol
Thrombozytopenie
Hogrefe
Anämien
Oxidationsmittel
Gelenkfläche
Biologische Bedeutung
Zytosol
Resuscitation
Astrozyten
Phagen
Trivialnamen
Decarboxylierung
Peptid
Zytoplasma
Lebensmittelzusatzstoff
Nephropathie
Proteingruppe
.H
Obstet
Septum
Neoplasie
Formeleinheiten
Blasonierung
Phloem
Diakonissenanstalt
Chloraniline
Natriumsalz
Zoonose
synaptischen Spalt
.Fachinformation
Chlorwasserstoff
Wochenschr
Pharmacol
Oxidationsstufe
Radioiodtherapie
Truncus
Physiol
Origanum
Vena
Phagozytose
Richtungsbezeichnungen
Depolarisation
Pharmakodynamik
Molekülmasse
HCl
Fruchtknoten
Glomerulonephritis
Elementarzelle
Oct
Skelettmuskel
Stenose
Spinalanästhesie
Medizinische Mikrobiologie
Operatives
Inhibitor
Weichstrahlen
Nystagmus
Orthologe
Sommer-Paralympics
Gitterparametern
Allgemeinanästhesie
Trivialname
Kontraktur
Gluconeogenese
Cytosol
VitaminD3
G-Protein-gekoppelten
West-Syndrom
Autorenkürzel
Doppelbindung
Agonist
Membranpotential
Homöostase
Metabolit
Systema
konzentrierter Schwefelsäure
Läsion

Chromosomenzahl
Untergattung
DNA-Polymerase
Klinische Angabe
Urban&FischerVerlag
Afferenzen
Ribosomen
Balsam-Tanne
Stoffwechselweg
Lokalanästhetika
wässriger Lösung
Erstbeschreiber
Phospholipid
IgE-Antikörper
Mineralklasse
Flagellen
Normaldruck
Reduktionsmittel
Krebsimmuntherapie
Filtrationsrate
Plattenepithel
Kofaktor
Stadieneinteilung
Echokardiografie
Golgi-Apparat
Arzneiform
Medizinische Bedeutung
Hohlvene
.Pressemitteilung
Oxalacetat
Ammoniumverbindung
Propen
glatten Muskulatur
Epidemiol
unabhängiges Informationsangebot
Alveolen
.C
Lumbalpunktion
Hänge-Birke
Schattauer
ATP-Synthase
Rezidive
Natriumcarbonat
Stroma
Medulla
botanisches Autorenkürzel
Ther
Metabolisierung
Pankreatitis
Neuroanatomie
Gefahrgutverordnung
Auskultation
S3-Leitlinie
Nukleotiden
e .a.
Reagenz
klinische Bild
Anion
Aszites
Nävus
zahlreiche Generikum
Krankheitsassoziiertes
Acetat
Stw.
Zystennieren
Keratinozyten
Hernie
spezielle Pharmakologie
Endozytose
Pharmakologische Eigenschaft
Hydroxygruppen
Differentialdiagnostisch
Erythrozyt
Alphateilchen
Treponema
Psychosozial-Verlag
Dihydrat
AV-Knoten
Hämochromatose
Alternativhypothese
Toluol
Calciumionen
Phosphatase
Coffein
Epithelien
Chromatiden
Differentialdiagnose /DD/
Proteobakterien
Differenzialdiagnose
Abduktion
kristallines Pulver
Metab
Pharmazeutische Zeitung
Leitsymptom
Taxon_Ausnahme
Ethylenoxid
Zündtemperatur
ICD-10
Epithel
Zytokin
anorganischen Chemie
Cytochrom
Hyperplasie
Glucose-6-phosphat
Histologisch
Intelligenzminderung
Ligand
HIV-1
Parathormon
Gefahrstoffverordnung
Orbitopathie
Konformation
Epoetin
Virusspezies
Kiemenbogen
AWMF-Registernummer
Pathobiochemie
Taijiquan
Medicin
Adenom
Rothalbmond-Gesellschaften
ZNS
explosionsfähiges Gemisch
Stoffklasse
Proteobacteria
Erstbeschreibung
Sekundärer
Isomerengemisch
Hartstrahlen
Monohydrat
Fibroblasten
Atrophie

Artikel
Ovulation
Konzentrationsgefälle
Embryogenese
Körperhöhlen
Hydrolyse
chemisches Element
Regelmäßige Veranstaltung
Lobus
neutrophilen Granulozyten
Histologie
Wasserstoffbrückenbindungen
Endothel
Pankreas
pathologische Anatomie
Akkommodation
Myopathie
Normalbedingungen
Humoralpathologie
Bisamäpfel
Siliciumdioxid
Temperaturklasse
Interleukine
Endothelien
Phosphorsäureester
äußeren Augenmuskel
wässrige Lösung
Klitorisvorhaut
unteren Extremität
Calcium-Ionen
Fibrin
Therapieoption
Herpes-simplex-Viren
Dikötter
Gastroenterol
Natriumverbindung
Protozoen
Orphan-Arzneimittel
Anilin
Gallensäuren
Elementsymbol
Gefäßpflanzen
Podozyten
Peritonitis
Methionin
NADH
Granulom
Virale
Nervi
Giftig
Hepatozyten
Peritoneum
Seitenkette
TNM-Klassifikation
Arterielle
Antimykotika
Klassifikators
farblose Kristalle
Glucocorticoiden
höchste Symmetrie
Organikum
Lichen
Amblyopie
Syn
Prodrug
kristalliner Feststoff
Biomembran
Epub
Arteriae
Gastroenteritis
Genort
Endplatte
Dampfdruckfunktion
Geleitzellen
Psychiatrie-Verlag
Vakuolen
Malformationen
Prostatahyperplasie
Urämie
Hemmstoffe
Permeabilität
Mediastinum
Vesikeln
Interstitium
Bulbus
Polymorphismus
Stoffgemisch
Virionen
Arzneiformen
Transgen
Carbonsäure
Volume
Handelsnamen
Atommasse
weitere Parameter
Kernteilung
Ossifikation
Aminosäuresequenz
Duodenum
Aminen
Hydrierung
Klassifikator
Mohshärte
Hyperkalzämie
Retikulum
Gruppenpsychotherapie
Dura
Carboxygruppe
Hypoglykämie
Massenanteil
NADPH
Gefahrstoff
Axons
Pharmakokinetik
Caesium
Pleura
G-Protein-gekoppelte
Betazerfall
Thrombozyten
Basalganglien
Hörschwelle
Wasseralfingen
Motoneurone
Siebröhren
Infektiöse
Review
Arzneibuch
Plastiden
Endokarditis

Zilien
klinischen Symptom
Syndrome
UN-Nummer
Leukopenie
Kristallwasser
Chondrozyten
Taxa
MP5
Glycolyse
Erregernachweis
Antioxidationsmittel
Hyperopie
Ileus
Natriumhydrogencarbonat
Metformin
Hrsg.
Elektronenkonfiguration
Isomere
Extrauteringravidität
DNA-Replikation
untere Explosionsgrenze
Sporenpulver
Differentialdiagnostik
Innere Medizin
Nukleotid
Arteriolen
Liquorraum
ableitenden Harnwege
Phagozyten
Ortsauflösung
Braillezeichen
Grenzfläche
MolaresVolumen
S2-Leitlinie
Wirtszelle
Rückresorption
Circulation
Hirnmetastasen
g/mol
symptomatischen Behandlung
Rezidiven
β-Carotin

Ärzte-Verlag
physiologischen Bedingung
IgE
Nuklid
Falsch-Positiv-Rate
Mykosen
Oxidationsmitteln
Augenabstand
Lichtmikroskopie
Deutsche Ärzte-Verlag
Retardierung
Arzneistoffes
β-Zellen
Huthaut
Embryologie
Fibrinogen
PubMed
Bupropion
Gonaden
Betastrahlung
Miktion
Hilfsmittelverzeichnis
molare Masse
Interleukin-2
Anaerobier
Calvin-Zyklus
C3-Pflanzen
Splicing
Eur J
Lactat
Kinderkardiologie
Transkriptionsfaktor
Geographische Lage
Efferenzen
Verdampfungswärme
Rotkreuz-Gesellschaften
Glycin
Calciumantagonisten
Patients
Neuroophthalmologie
Konjunktivitis
Versorgungsstufe
obere Explosionsgrenze

Zentralblatt
Standardbedingungen
Kolophonium
Pathophysiologie
Zytologie
Spezifität
Citrat
Kohlenstoffmonoxid
Harnstoffzyklus
Wasserlöslichkeit
Konformationsänderung
Ribose
Tubulus
Bindungsstelle
eukaryotischen Zelle
Glycerintrinitrat
Gyrus
Schallkopf
Hämodialyse
Epithels
Phenobarbital
Supination
Lokalisationen
Vasokonstriktion
Vakuole
Pflanzenernährung
Klinisches
ISSN
Papille
Gesäßmuskel
oraler Gabe
FSGS
Gutartige
Parese
Hypoglykämien
Nährmedium
Hallenweltmeisterschaften
Schamhaarentfernung
Hyperpolarisation
Parasitose
Ohrenheilkunde
Bauchfells
Ribosom

Haplogruppe
MP5K
Interleukin-12
Sprungbein
Europäischen Arzneibuch
Exozytose
Fluorchinolone
deutscher Chirurg
Belegzellen
Benzaldehyd
Prostaglandine
Paresen
OH-Gruppe
Venerologie
Ethin
Transversalebene
digitalisierte Fassung
Einklemmung
T-Zelle
Aldehyd
Entzündungsmediatoren
Orobanche
Aktin
Hirnnerv
Sauerstoffspezies
Sonstige Information
Fructosemalabsorption
Arzneimittelzulassung
Gruppenanalyse
diverse Generikum
krautige Pflanze
Oxide
Chromatin
Antikoagulantien
Hypokaliämie
Motorcortex
Wirkstoffgruppe

hervorragender Ärzte
Gastrointestinaltrakt
Parenchym
Transplantates
Infection
Reduktionsäquivalente
Aortenbogen
Polyploidie
Wirkungseintritt
Biomembranen
Pneumothorax
Medicine
Vorwölbung
Metaboliten
Osteotomie
Chloroplasten
Glykosylierung
schweizerischen Bundesamt
farblose Flüssigkeit
Zellproliferation
Propionsäure
Gv-Pflanzen
Arzneistoffe
Anaesthesist
Pränatalmedizin
α-Untereinheit
C4-Pflanzen
Subarachnoidalblutung
Disulfidbrücken
Zirconium
Komorbiditäten
Kaliumhydroxid
Anheftung
Formatio
Knochenfischen
Viruspartikel
Alphazerfall

Bilirubin
Zirkumzision
Oxid
Gemeindegliederung
Appendizitis
Antiphlogistika
Gesundheitshinweis
Pentosephosphatweg
erweiterte Auflage
Rezidivrate
5'-Ende
Hydroxylierung
Fissura
CAM-Pflanzen
Schieloperation
Feinbau
Antikoagulation
Sorbinsäure
Elektronenakzeptor
Basismaßnahmen
Elsevier
Malabsorption
Metamizol
Makrophagen
Introns
Hexahydrat
Citronensäure
NAD+
Epiphyse
Skelettmuskeln
Hals-Nasen-Ohren-Heilkunde
Adenokarzinom
Elektronegativität
Spinalnerven
Metabolite
Vitalkapazität
Tyrosin